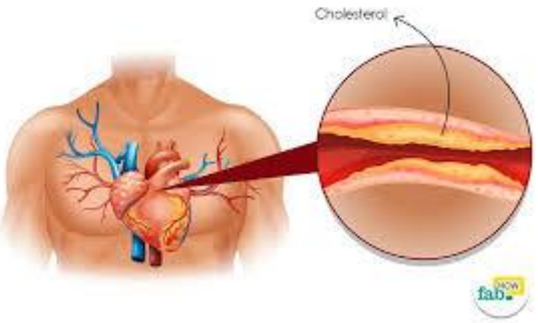
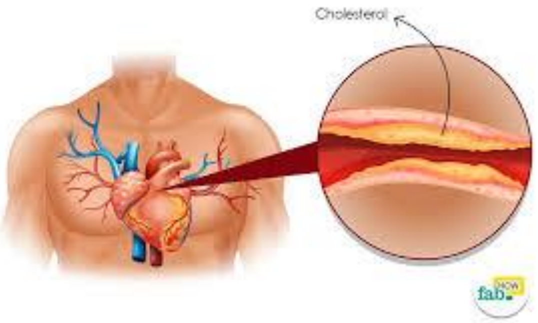
Know Your Data- Python

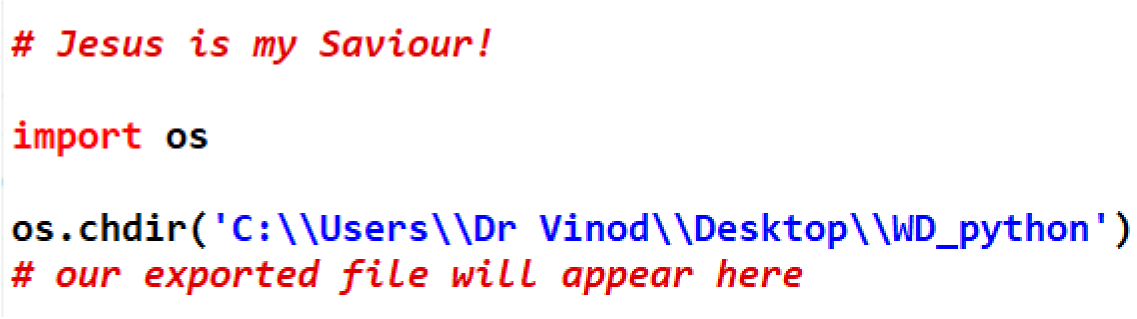
Data Sets: **cs2m** & **grades**



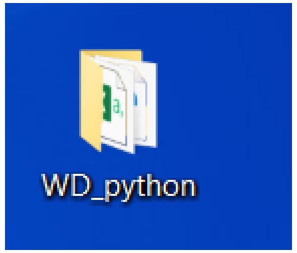
Data

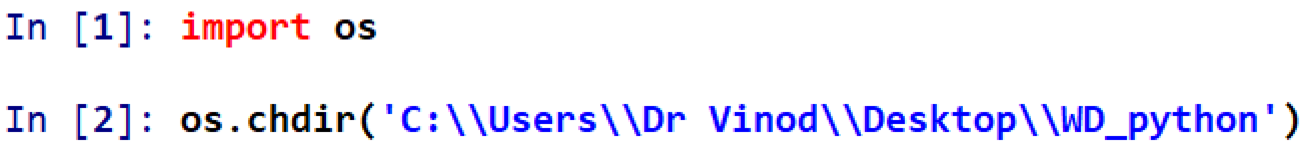


Creating Working Directory



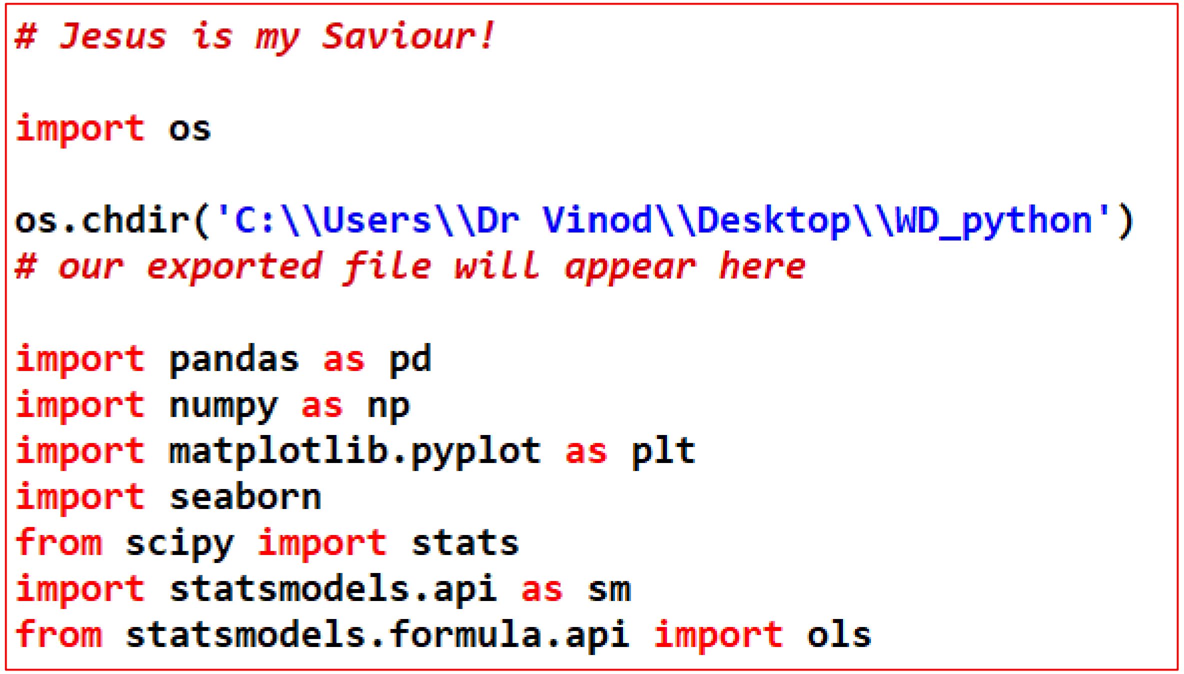
```
# Jesus is my Saviour!

import os

os.chdir('C:\\Users\\Dr Vinod\\Desktop\\WD_python')
# our exported file will appear here
```



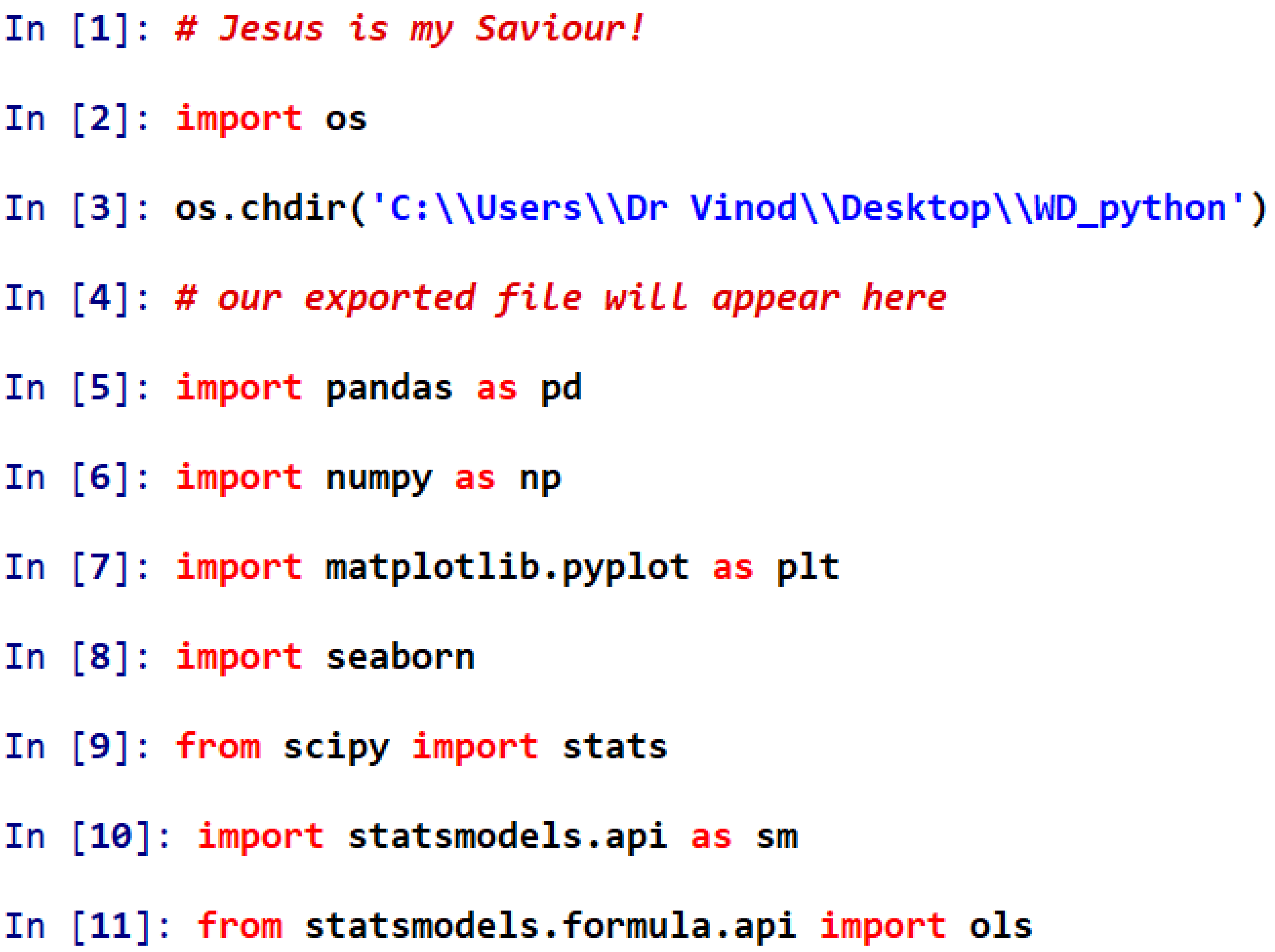


```
In [1]: import os

In [2]: os.chdir('C:\\Users\\Dr Vinod\\Desktop\\WD_python')
```

Necessary Libraries

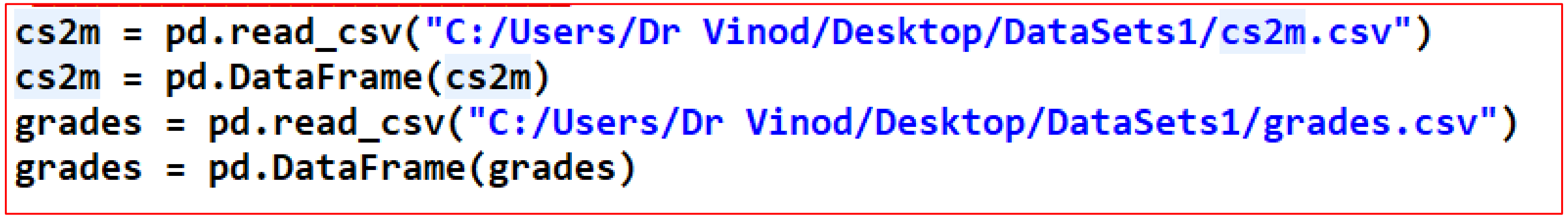


```
# Jesus is my Saviour!

import os

os.chdir('C:\\Users\\Dr Vinod\\Desktop\\WD_python')
# our exported file will appear here

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn
from scipy import stats
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

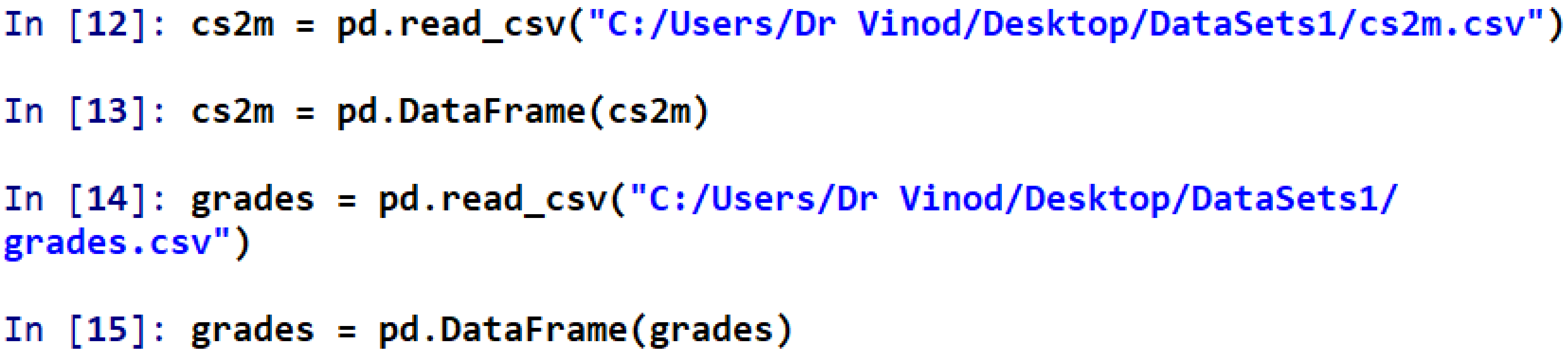


```
In [1]: # Jesus is my Saviour!

In [2]: import os

In [3]: os.chdir('C:\\Users\\Dr Vinod\\Desktop\\WD_python')

In [4]: # our exported file will appear here

In [5]: import pandas as pd

In [6]: import numpy as np

In [7]: import matplotlib.pyplot as plt

In [8]: import seaborn

In [9]: from scipy import stats

In [10]: import statsmodels.api as sm

In [11]: from statsmodels.formula.api import ols
```

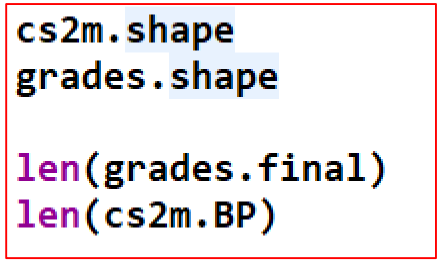
Import Data



```
cs2m = pd.read_csv("C:/Users/Dr Vinod/Desktop/DataSets1/cs2m.csv")
cs2m = pd.DataFrame(cs2m)
grades = pd.read_csv("C:/Users/Dr Vinod/Desktop/DataSets1/grades.csv")
grades = pd.DataFrame(grades)
```

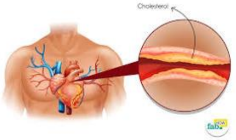


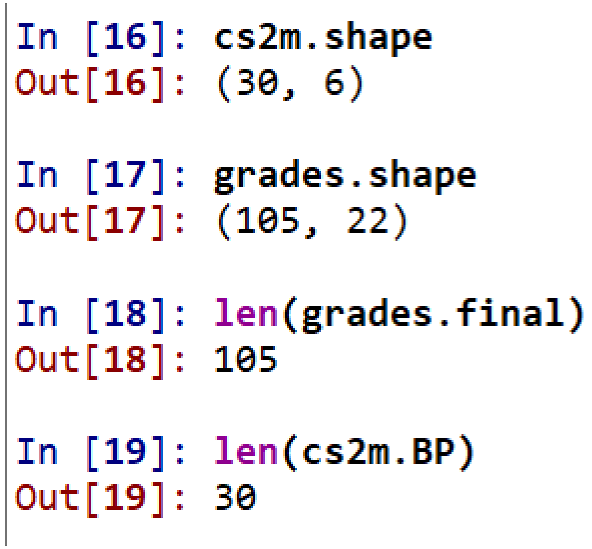
```
In [12]: cs2m = pd.read_csv("C:/Users/Dr Vinod/Desktop/DataSets1/cs2m.csv")

In [13]: cs2m = pd.DataFrame(cs2m)

In [14]: grades = pd.read_csv("C:/Users/Dr Vinod/Desktop/DataSets1/
grades.csv")

In [15]: grades = pd.DataFrame(grades)
```



```
cs2m.shape
grades.shape

len(grades.final)
len(cs2m.BP)
```

Appearance of Data

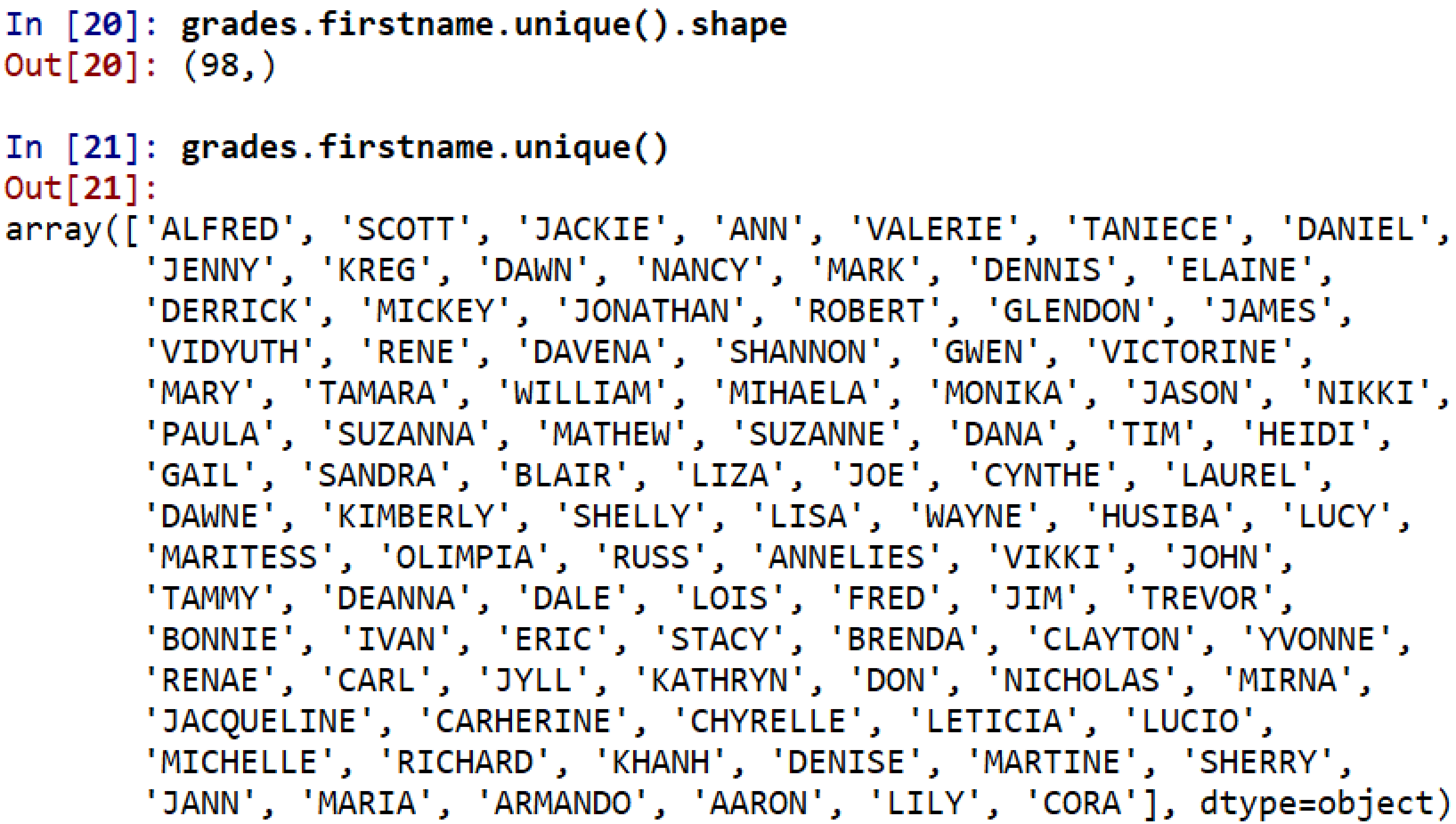




```
In [16]: cs2m.shape
Out[16]: (30, 6)

In [17]: grades.shape
Out[17]: (105, 22)

In [18]: len(grades.final)
Out[18]: 105

In [19]: len(cs2m.BP)
Out[19]: 30
```

```
grades.firstname.unique().shape
grades.firstname.unique()
```

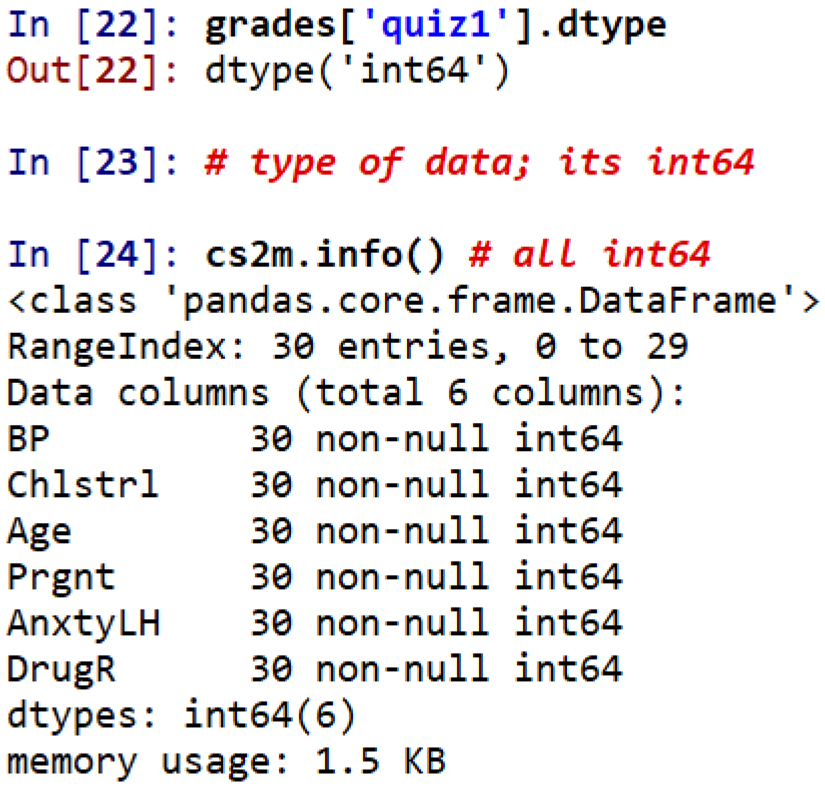
First Name



```
In [20]: grades.firstname.unique().shape
Out[20]: (98,)

In [21]: grades.firstname.unique()
Out[21]:
array(['ALFRED', 'SCOTT', 'JACKIE', 'ANN', 'VALERIE', 'TANIECE', 'DANIEL',
       'JENNY', 'KREG', 'DAWN', 'NANCY', 'MARK', 'DENNIS', 'ELAINE',
       'DERRICK', 'MICKEY', 'JONATHAN', 'ROBERT', 'GLENDON', 'JAMES',
       'VIDYUTH', 'RENE', 'DAVENA', 'SHANNON', 'GWEN', 'VICTORINE',
       'MARY', 'TAMARA', 'WILLIAM', 'MIHAELA', 'MONIKA', 'JASON', 'NIKKI',
       'PAULA', 'SUZANNA', 'MATHEW', 'SUZANNE', 'DANA', 'TIM', 'HEIDI',
       'GAIL', 'SANDRA', 'BLAIR', 'LIZA', 'JOE', 'CYNTHE', 'LAUREL',
       'DAWNE', 'KIMBERLY', 'SHELLY', 'LISA', 'WAYNE', 'HUSIBA', 'LUCY',
       'MARITESS', 'OLIMPIA', 'RUSS', 'ANNELIES', 'VIKKI', 'JOHN',
       'TAMMY', 'DEANNA', 'DALE', 'LOIS', 'FRED', 'JIM', 'TREVOR',
       'BONNIE', 'IVAN', 'ERIC', 'STACY', 'BRENDA', 'CLAYTON', 'YVONNE',
       'RENAE', 'CARL', 'JYLL', 'KATHRYN', 'DON', 'NICHOLAS', 'MIRNA',
       'JACQUELINE', 'CARHERINE', 'CHYRELLE', 'LETICIA', 'LUCIO',
       'MICHELLE', 'RICHARD', 'KHANH', 'DENISE', 'MARTINE', 'SHERRY',
       'JANN', 'MARIA', 'ARMANDO', 'AARON', 'LILY', 'CORA'], dtype=object)
```

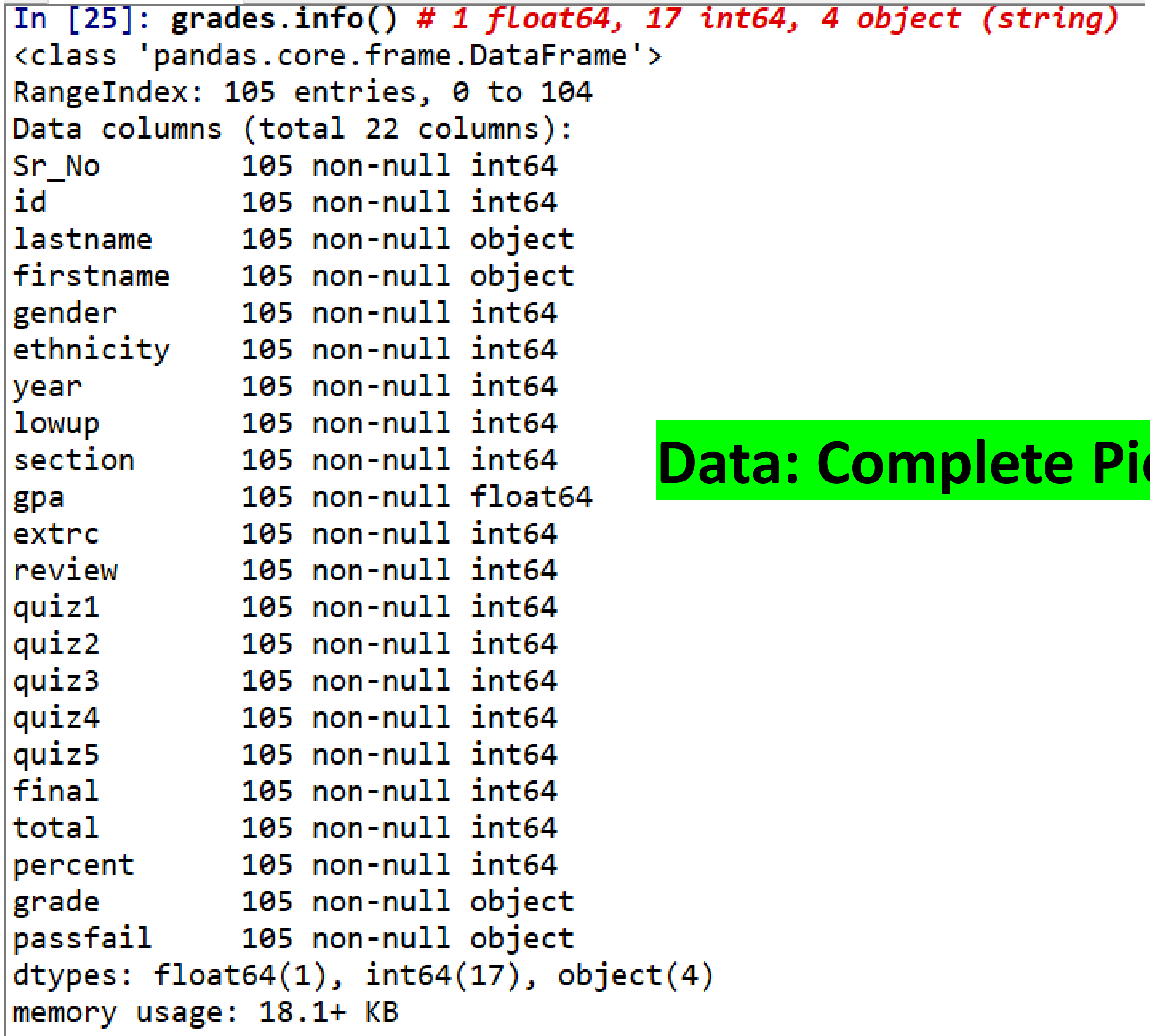
Data Type

```
grades['quiz1'].dtype
# type of data; its int64

cs2m.info() # all int64
```



```
In [22]: grades['quiz1'].dtype
Out[22]: dtype('int64')

In [23]: # type of data; its int64

In [24]: cs2m.info() # all int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 6 columns):
BP          30 non-null int64
Chlstrl     30 non-null int64
Age         30 non-null int64
Prgnt       30 non-null int64
AnxtyLH     30 non-null int64
DrugR       30 non-null int64
dtypes: int64(6)
memory usage: 1.5 KB
```

```
grades.info() # 1 float64, 17 int64, 4 object (string)
```

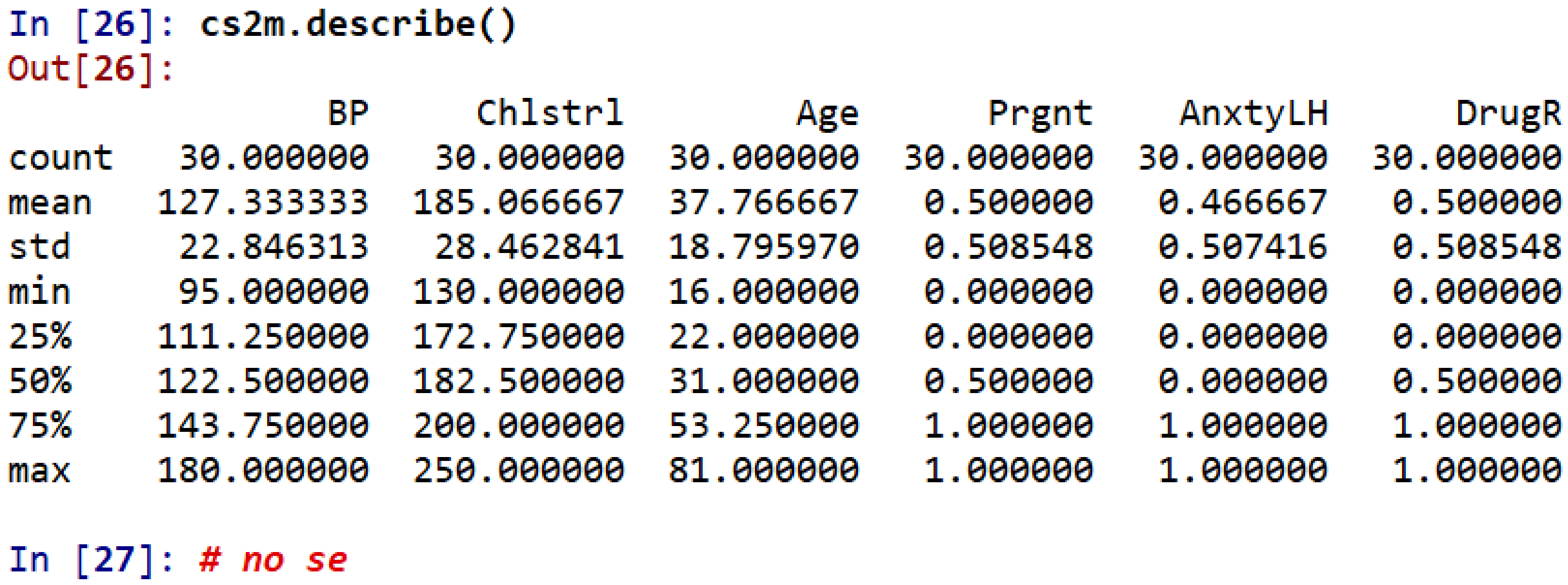


```
In [25]: grades.info() # 1 float64, 17 int64, 4 object (string)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 105 entries, 0 to 104
Data columns (total 22 columns):
Sr_No        105 non-null int64
id           105 non-null int64
lastname     105 non-null object
firstname    105 non-null object
gender       105 non-null int64
ethnicity    105 non-null int64
year         105 non-null int64
lowup        105 non-null int64
section      105 non-null int64
gpa          105 non-null float64
extrc        105 non-null int64
review       105 non-null int64
quiz1        105 non-null int64
quiz2        105 non-null int64
quiz3        105 non-null int64
quiz4        105 non-null int64
quiz5        105 non-null int64
final        105 non-null int64
total        105 non-null int64
percent      105 non-null int64
grade        105 non-null object
passfail     105 non-null object
dtypes: float64(1), int64(17), object(4)
memory usage: 18.1+ KB
```

Data: Complete Picture

```
cs2m.describe()
# no se
```

Data: Complete Picture



```
In [26]: cs2m.describe()
Out[26]: 
               BP     Chlstrl        Age      Prgnt    AnxtyLH      DrugR
count   30.000000   30.000000  30.000000  30.000000  30.000000  30.000000
mean   127.333333  185.066667  37.766667   0.500000   0.466667   0.500000
std     22.846313   28.462841  18.795970   0.508548   0.507416   0.508548
min     95.000000  130.000000  16.000000   0.000000   0.000000   0.000000
25%    111.250000  172.750000  22.000000   0.000000   0.000000   0.000000
50%    122.500000  182.500000  31.000000   0.500000   0.000000   0.500000
75%    143.750000  200.000000  53.250000   1.000000   1.000000   1.000000
max    180.000000  250.000000  81.000000   1.000000   1.000000   1.000000

In [27]: # no se
```

```
cs2m['Age'].describe()

cs2m.Age.groupby(cs2m.Prgnt).describe()
```

Age versus Pregnancy

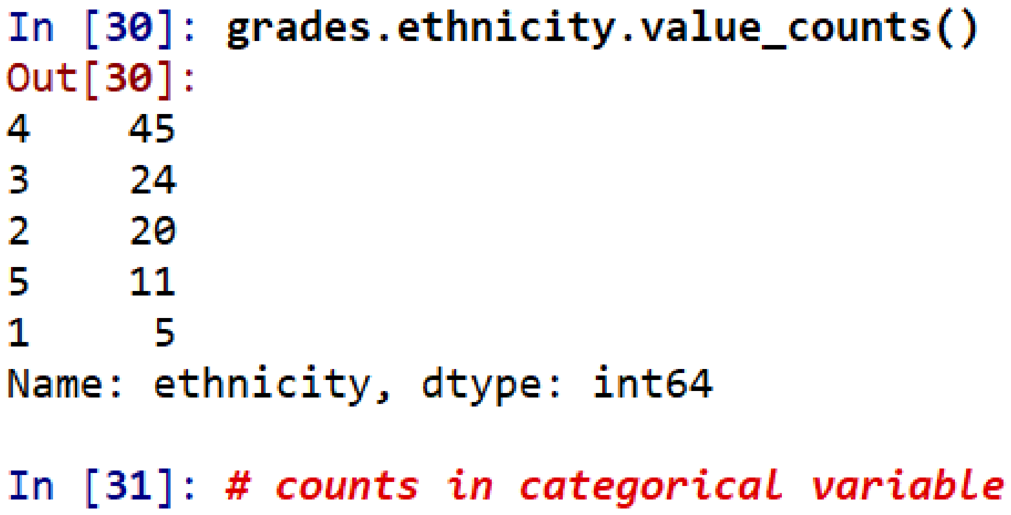
```
In [28]: cs2m['Age'].describe()
Out[28]:
count    30.000000
mean     37.766667
std      18.795970
min      16.000000
25%      22.000000
50%      31.000000
75%      53.250000
max      81.000000
Name: Age, dtype: float64
```

ɪrstock.com • 1074559082

```
In [29]: cs2m.Age.groupby(cs2m.Prgnt).describe()
Out[29]:
       count       mean        std   min   25%   50%   75%   max
Prgnt
0       15.0  48.000000  21.350811  16.0  30.5  56.0  62.0  81.0
1       15.0  27.533333   7.179999  18.0  20.5  29.0  31.0  40.0
```

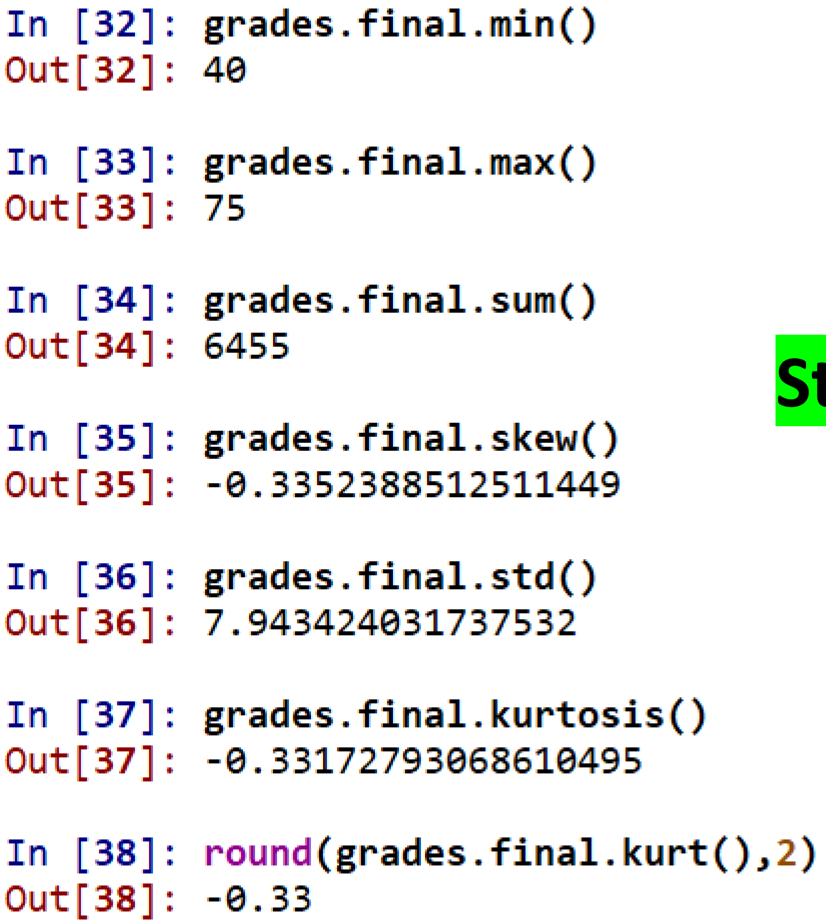
```
grades.ethnicity.value_counts()
# counts in categorical variable
```

Counts in Categorical Variable



```
In [30]: grades.ethnicity.value_counts()
Out[30]:
4    45
3    24
2    20
5    11
1     5
Name: ethnicity, dtype: int64

In [31]: # counts in categorical variable
```

```
grades.final.min()
grades.final.max()
grades.final.sum()
grades.final.skew()
grades.final.std()
grades.final.kurtosis()
round(grades.final.kurt(),2)
```

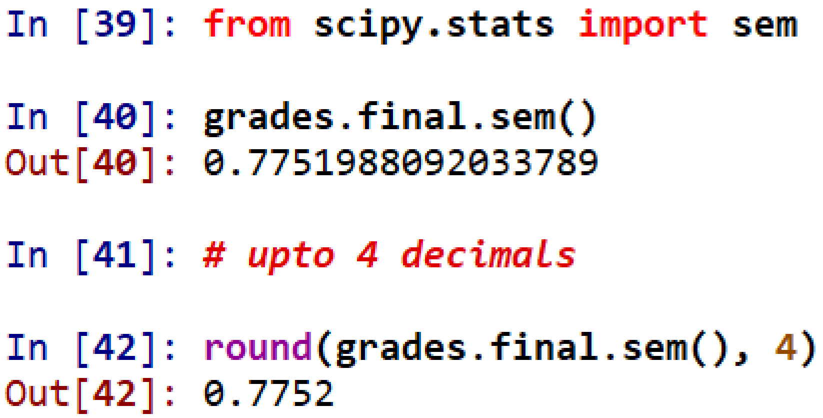


```
In [32]: grades.final.min()
Out[32]: 40

In [33]: grades.final.max()
Out[33]: 75

In [34]: grades.final.sum()
Out[34]: 6455

In [35]: grades.final.skew()
Out[35]: -0.335238851251149

In [36]: grades.final.std()
Out[36]: 7.94342403173753

In [37]: grades.final.kurtosis()
Out[37]: -0.331727930686104 95

In [38]: round(grades.final.kurt(),2)
Out[38]: -0.33
```

Statistics

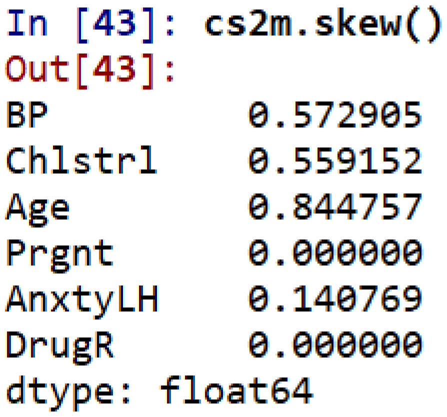
```
from scipy.stats import sem
grades.final.sem()

# upto 4 decimals
round(grades.final.sem(), 4)
```

Statistics

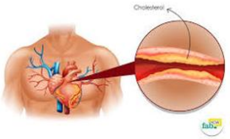


```
In [39]: from scipy.stats import sem

In [40]: grades.final.sem()
Out[40]: 0.7751988092033789

In [41]: # upto 4 decimals

In [42]: round(grades.final.sem(), 4)
Out[42]: 0.7752
```

```
cs2m.skew()
```

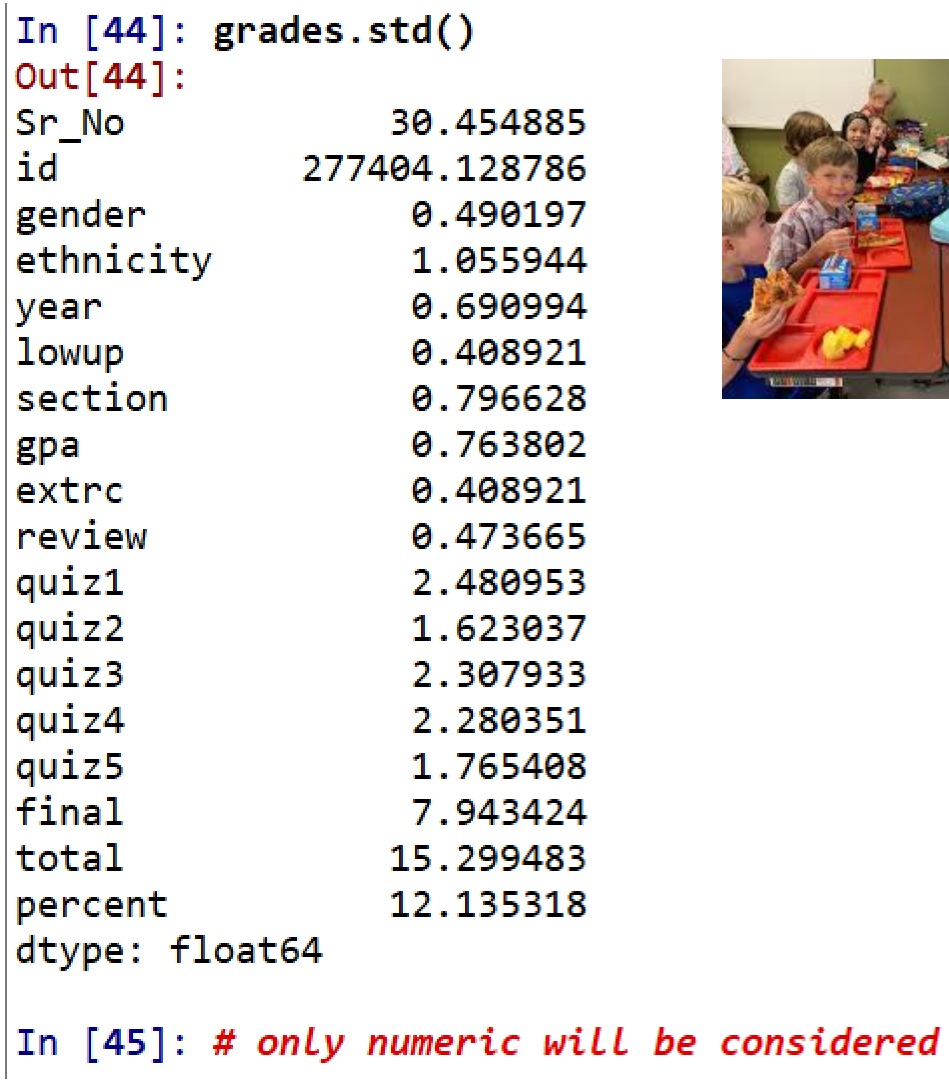


```
In [43]: cs2m.skew()
Out[43]:
BP         0.572905
Chlstrl    0.559152
Age        0.844757
Prgnt      0.000000
AnxtyLH    0.140769
DrugR      0.000000
dtype: float64
```



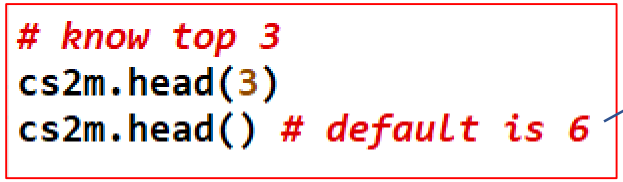
Statistics

```
grades.std()
# only numeric will be considered
```



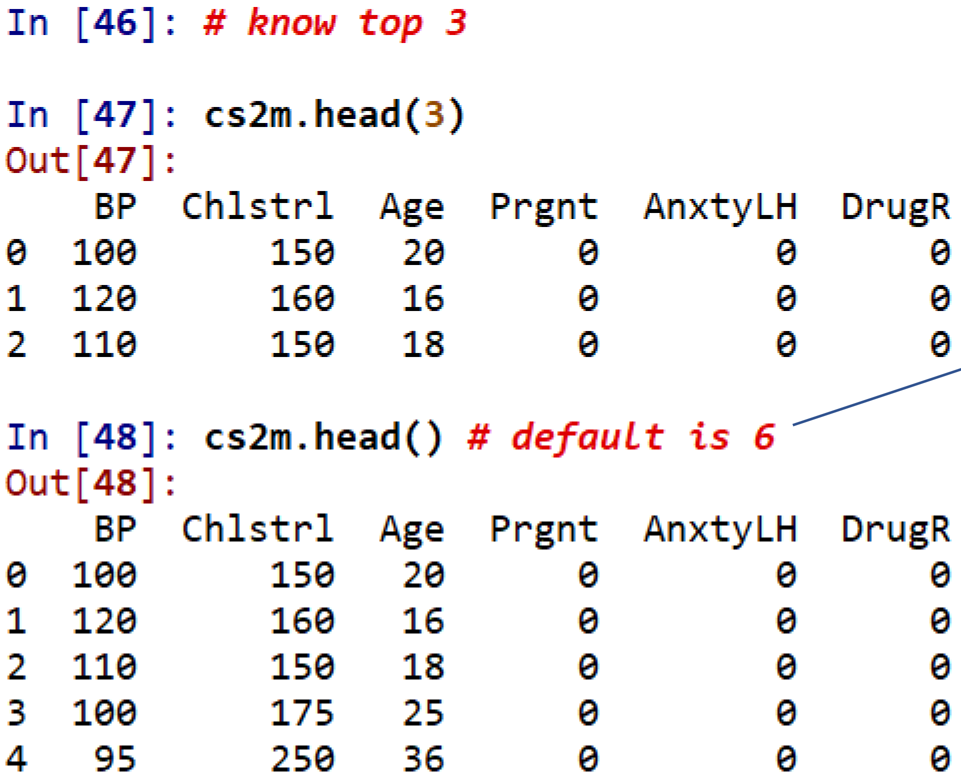
```
In [44]: grades.std()
Out[44]:
Sr_No            30.454885
id           277404.128786
gender            0.490197
ethnicity         1.055944
year              0.690994
lowup             0.408921
section           0.796628
gpa               0.763802
extrc             0.408921
review            0.473665
quiz1             2.480953
quiz2             1.623037
quiz3             2.307933
quiz4             2.280351
quiz5             1.765408
final             7.943424
total            15.299483
percent          12.135318
dtype: float64

In [45]: # only numeric will be considered
```

Top Rows



```
# know top 3
cs2m.head(3)
cs2m.head() # default is 6
```

Its 5

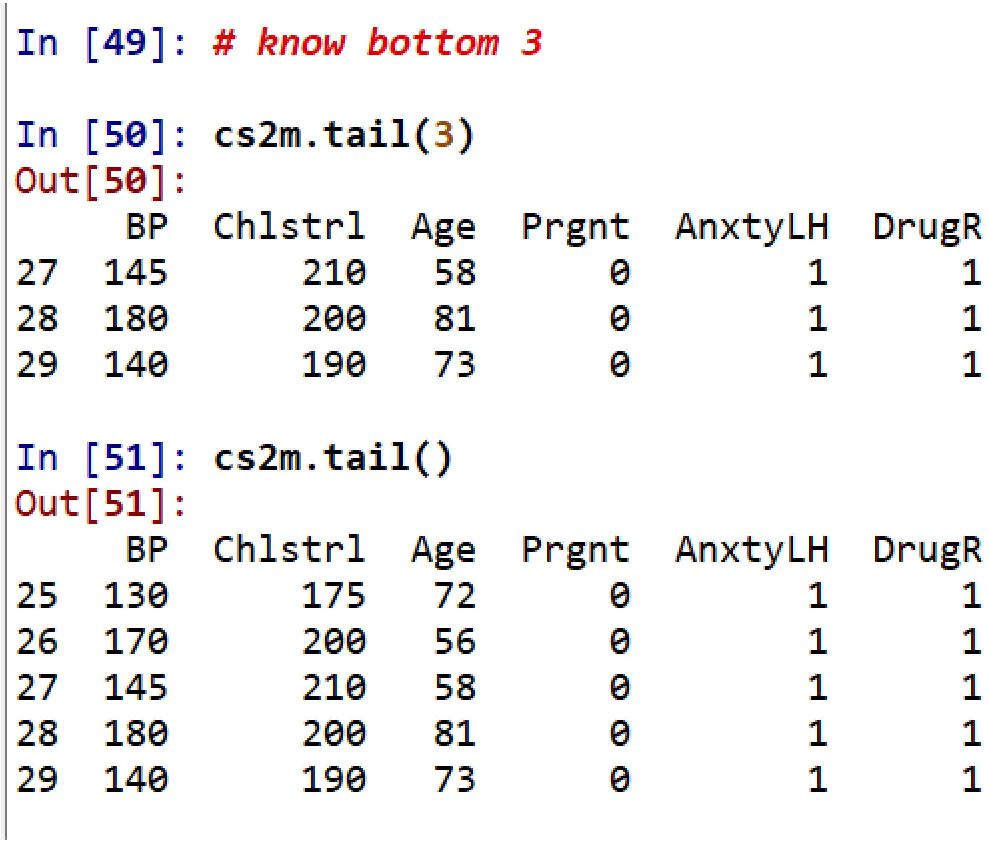


```
In [46]: # know top 3

In [47]: cs2m.head(3)
Out[47]:
    BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR
0  100      150   20      0        0      0
1  120      160   16      0        0      0
2  110      150   18      0        0      0

In [48]: cs2m.head() # default is 6
Out[48]:
    BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR
0  100      150   20      0        0      0
1  120      160   16      0        0      0
2  110      150   18      0        0      0
3  100      175   25      0        0      0
4   95      250   36      0        0      0
```

Its 5

Bottom Rows

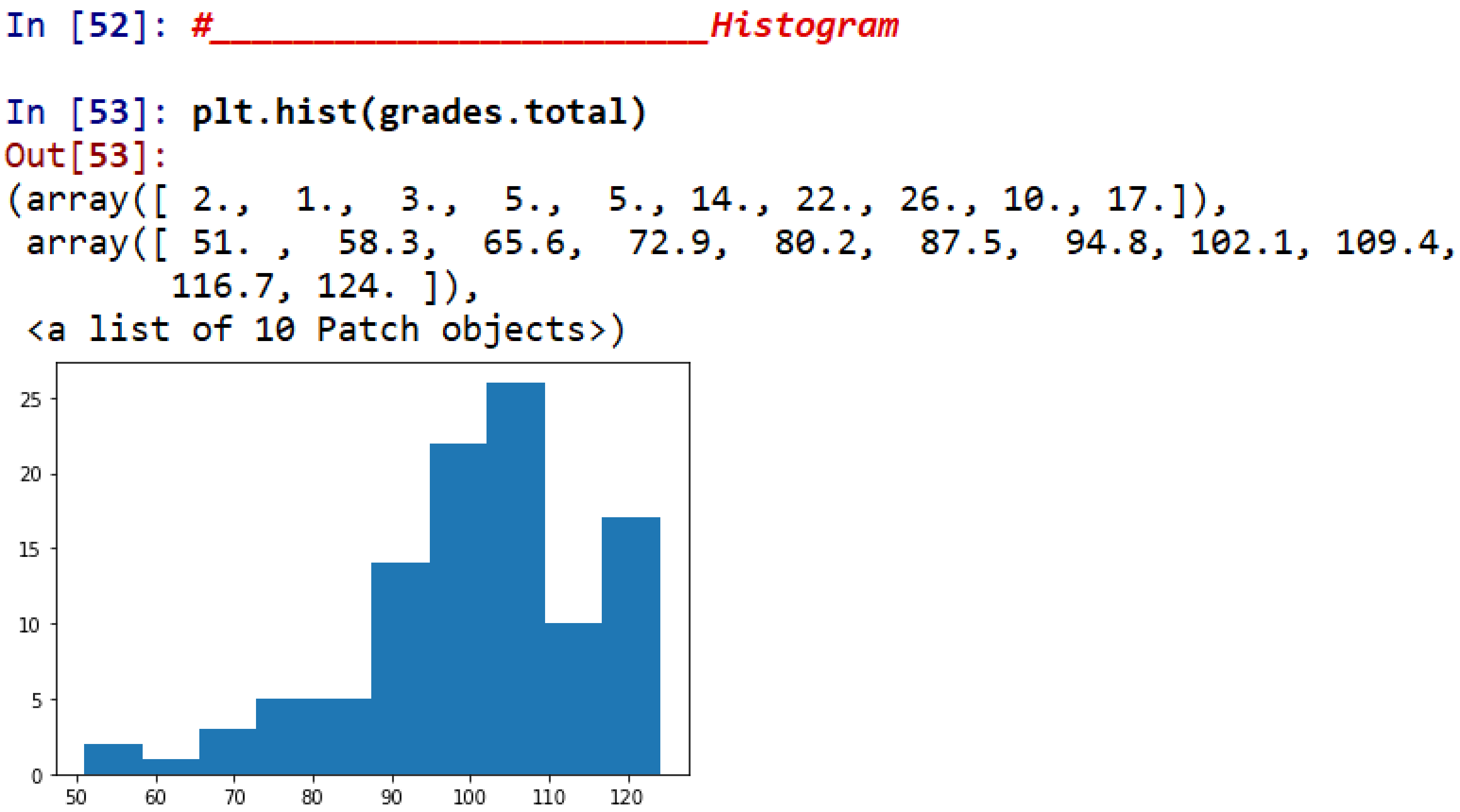
```
# know bottom 3
cs2m.tail(3)
cs2m.tail()
```



```
In [49]: # know bottom 3

In [50]: cs2m.tail(3)
Out[50]:
     BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR
27  145      210   58      0        1      1
28  180      200   81      0        1      1
29  140      190   73      0        1      1

In [51]: cs2m.tail()
Out[51]:
     BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR
25  130      175   72      0        1      1
26  170      200   56      0        1      1
27  145      210   58      0        1      1
28  180      200   81      0        1      1
29  140      190   73      0        1      1
```

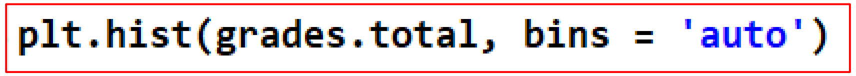
```
#________________________Histogram

plt.hist(grades.total)
```

Histogram

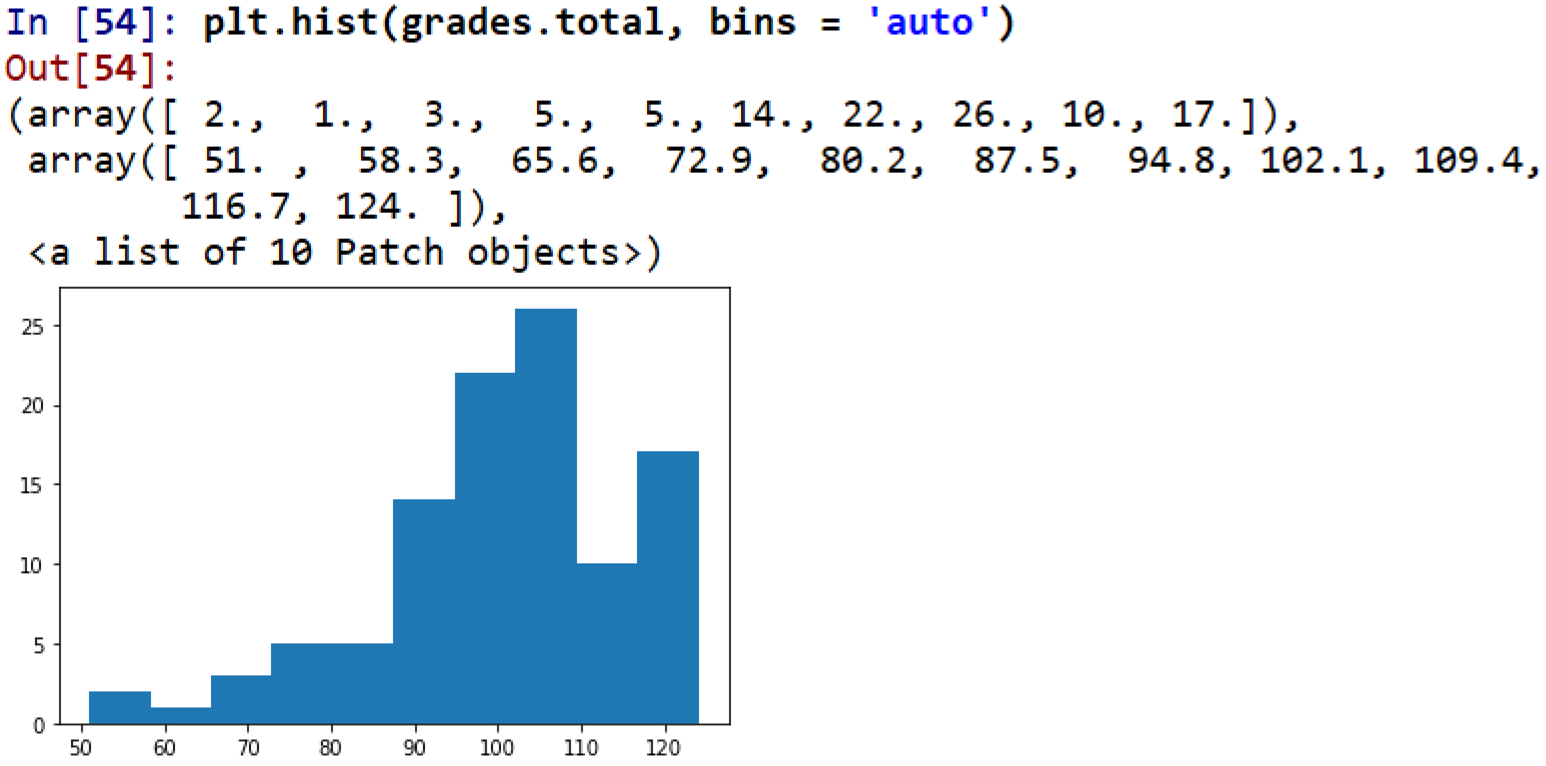


```
In [52]: #________________________Histogram

In [53]: plt.hist(grades.total)
Out[53]:
(array([ 2.,  1.,  3.,  5.,  5., 14., 22., 26., 10., 17.]),
 array([ 51. ,  58.3,  65.6,  72.9,  80.2,  87.5,  94.8, 102.1, 109.4,
        116.7, 124. ]),
 <a list of 10 Patch objects>)
```

Histogram



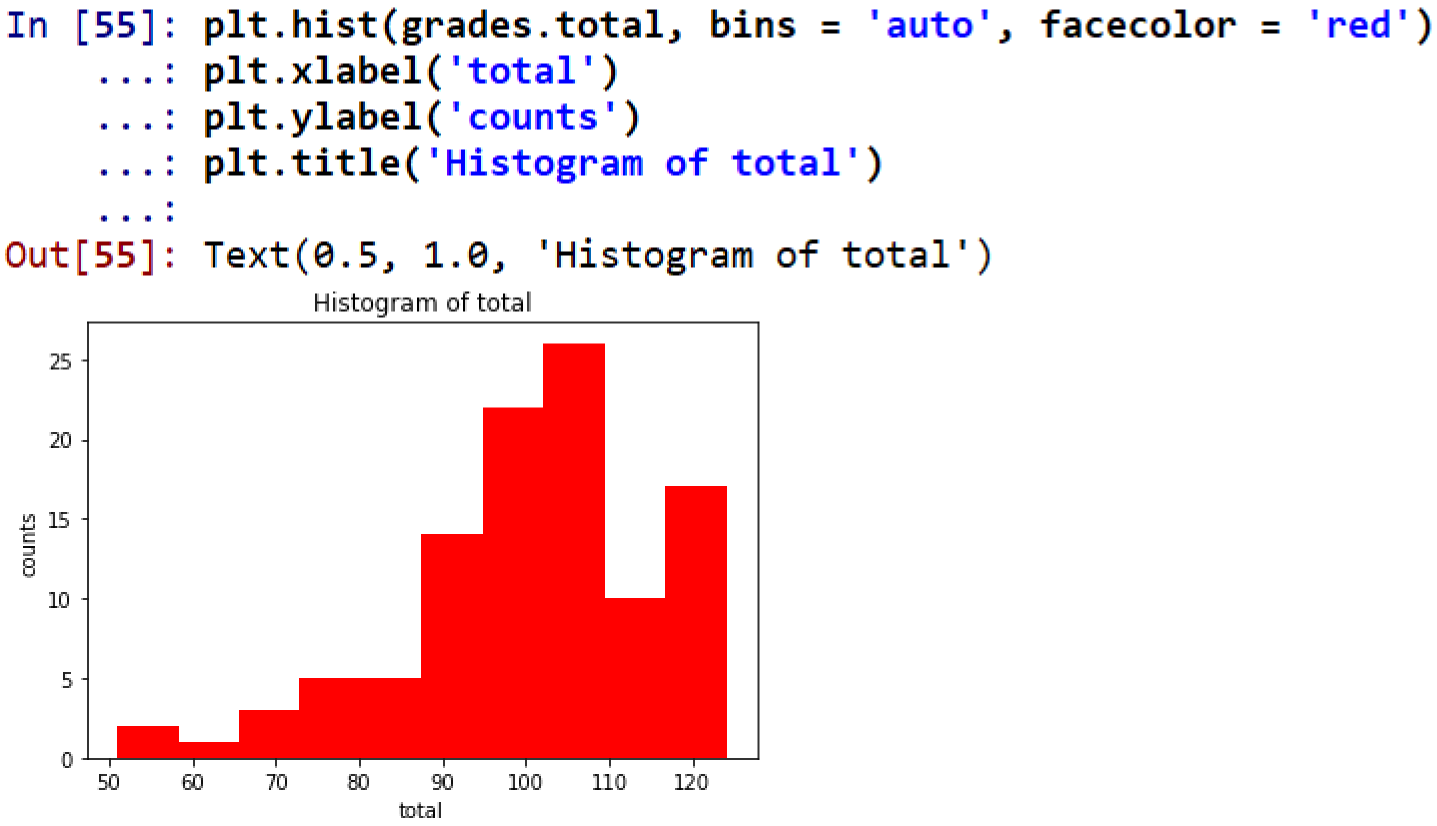
```
plt.hist(grades.total, bins = 'auto')
```



```
In [54]: plt.hist(grades.total, bins = 'auto')
Out[54]:
(array([ 2.,  1.,  3.,  5.,  5., 14., 22., 26., 10., 17.]),
 array([ 51. ,  58.3,  65.6,  72.9,  80.2,  87.5,  94.8, 102.1, 109.4,
        116.7, 124. ]),
 <a list of 10 Patch objects>)
```

```
# do all below 4 together
plt.hist(grades.total, bins = 'auto', facecolor = 'red')
plt.xlabel('total')
plt.ylabel('counts')
plt.title('Histogram of total')
```

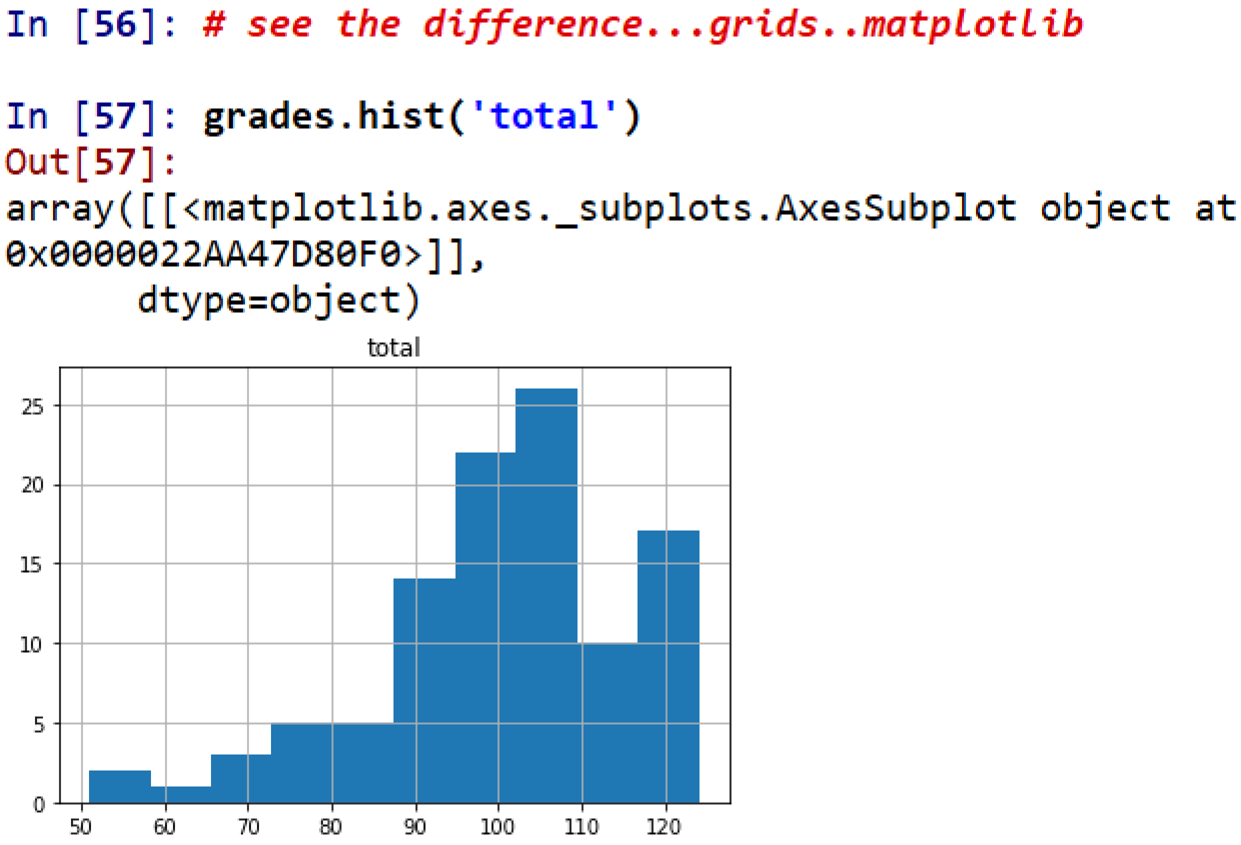
Histogram



```
In [55]: plt.hist(grades.total, bins = 'auto', facecolor = 'red')
    ...: plt.xlabel('total')
    ...: plt.ylabel('counts')
    ...: plt.title('Histogram of total')
    ...: 
Out[55]: Text(0.5, 1.0, 'Histogram of total')
```

```
# see the difference...grids..matplotlib
grades.hist('total')
```

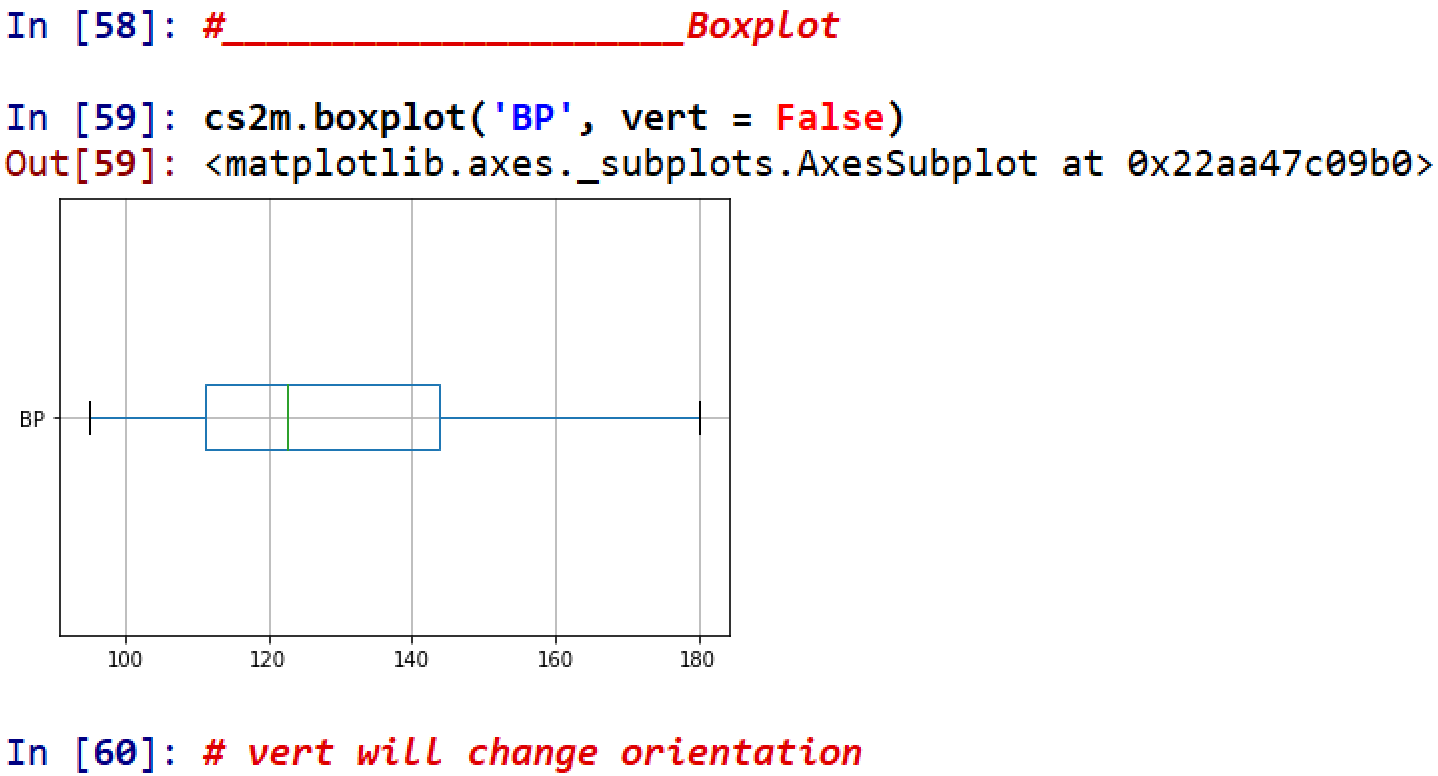
Histogram



```
In [56]: # see the difference...grids..matplotlib

In [57]: grades.hist('total')
Out[57]:
array([[<matplotlib.axes._subplots.AxesSubplot object at
0x0000022AA47D80F0>]],
      dtype=object)
```

Box Plot

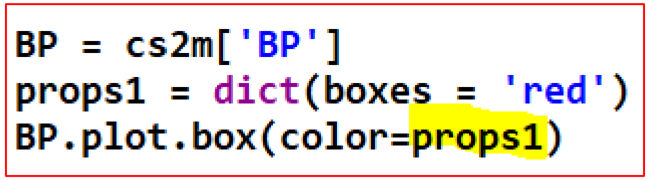
```
#____________________Boxplot

cs2m.boxplot('BP', vert = False)
# vert will change orientation
```



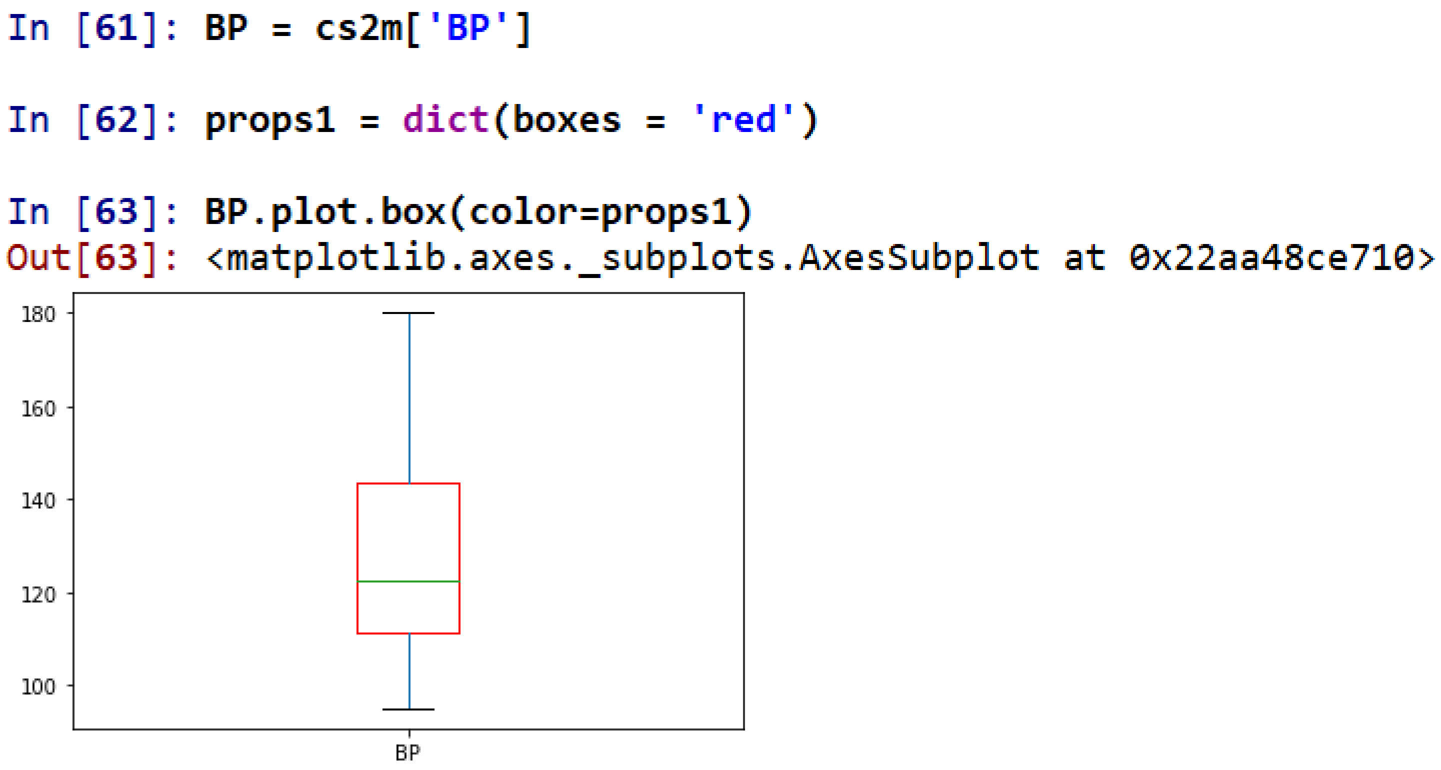
```
In [58]: #____________________Boxplot

In [59]: cs2m.boxplot('BP', vert = False)
Out[59]: <matplotlib.axes._subplots.AxesSubplot at 0x22aa47c09b0>
```

```
In [60]: # vert will change orientation
```

Box Plot

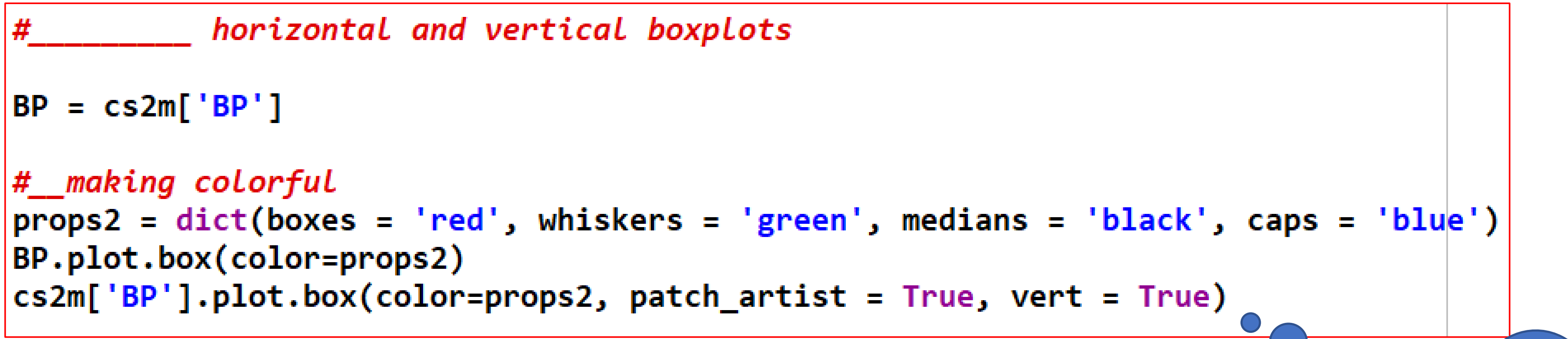


```
BP = cs2m['BP']
props1 = dict(boxes = 'red')
BP.plot.box(color=props1)
```



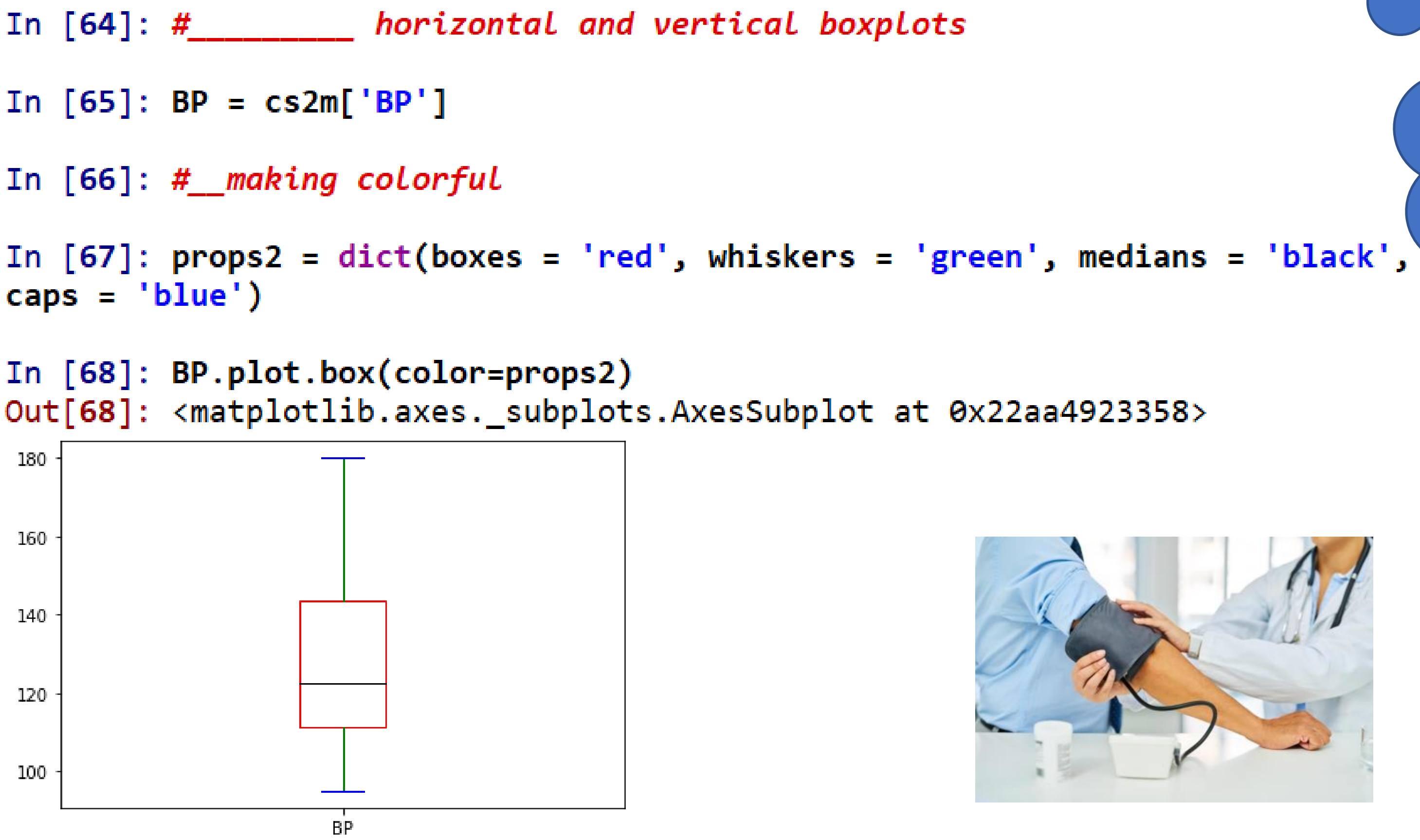
```
In [61]: BP = cs2m['BP']

In [62]: props1 = dict(boxes = 'red')

In [63]: BP.plot.box(color=props1)
Out[63]: <matplotlib.axes._subplots.AxesSubplot at 0x22aa48ce710>
```

Box Plot



```
#_________ horizontal and vertical boxplots

BP = cs2m['BP']

#__making colorful
props2 = dict(boxes = 'red', whiskers = 'green', medians = 'black', caps = 'blue')
BP.plot.box(color=props2)
cs2m['BP'].plot.box(color=props2, patch_artist = True, vert = True)
```

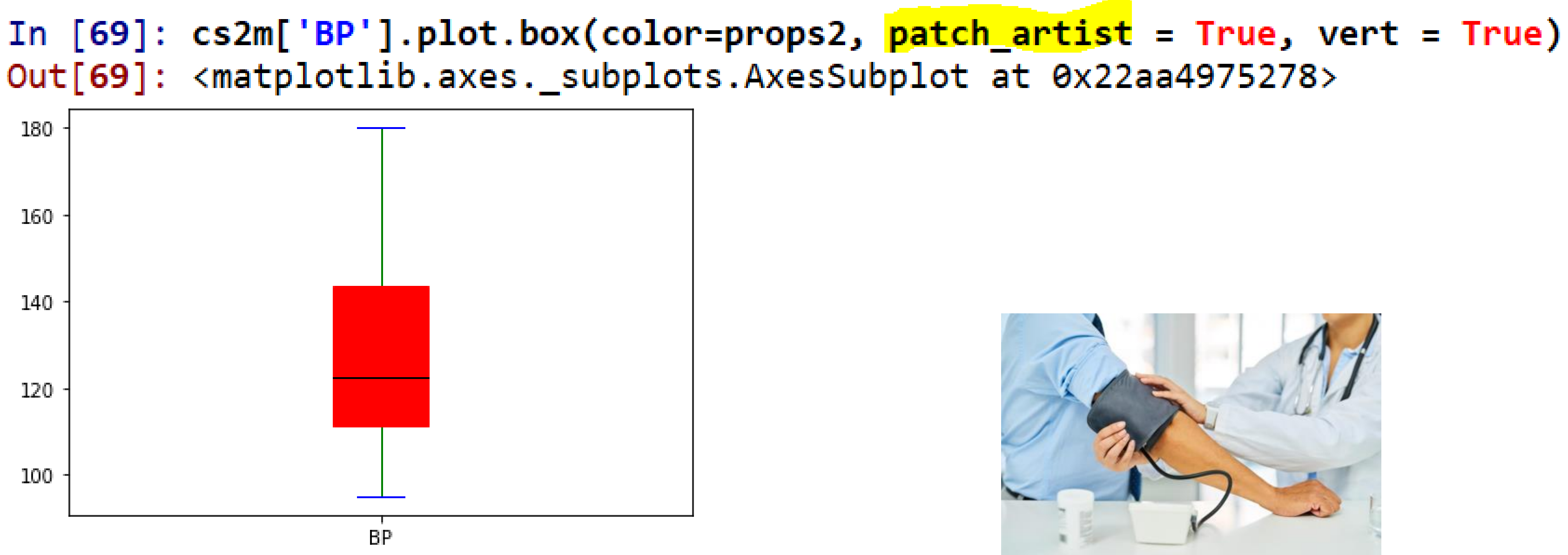
See the result of patch_artist = True in next slide



```
In [64]: #_________ horizontal and vertical boxplots

In [65]: BP = cs2m['BP']

In [66]: #__making colorful

In [67]: props2 = dict(boxes = 'red', whiskers = 'green', medians = 'black',
caps = 'blue')

In [68]: BP.plot.box(color=props2)
Out[68]: <matplotlib.axes._subplots.AxesSubplot at 0x22aa4923358>
```

Box Plot

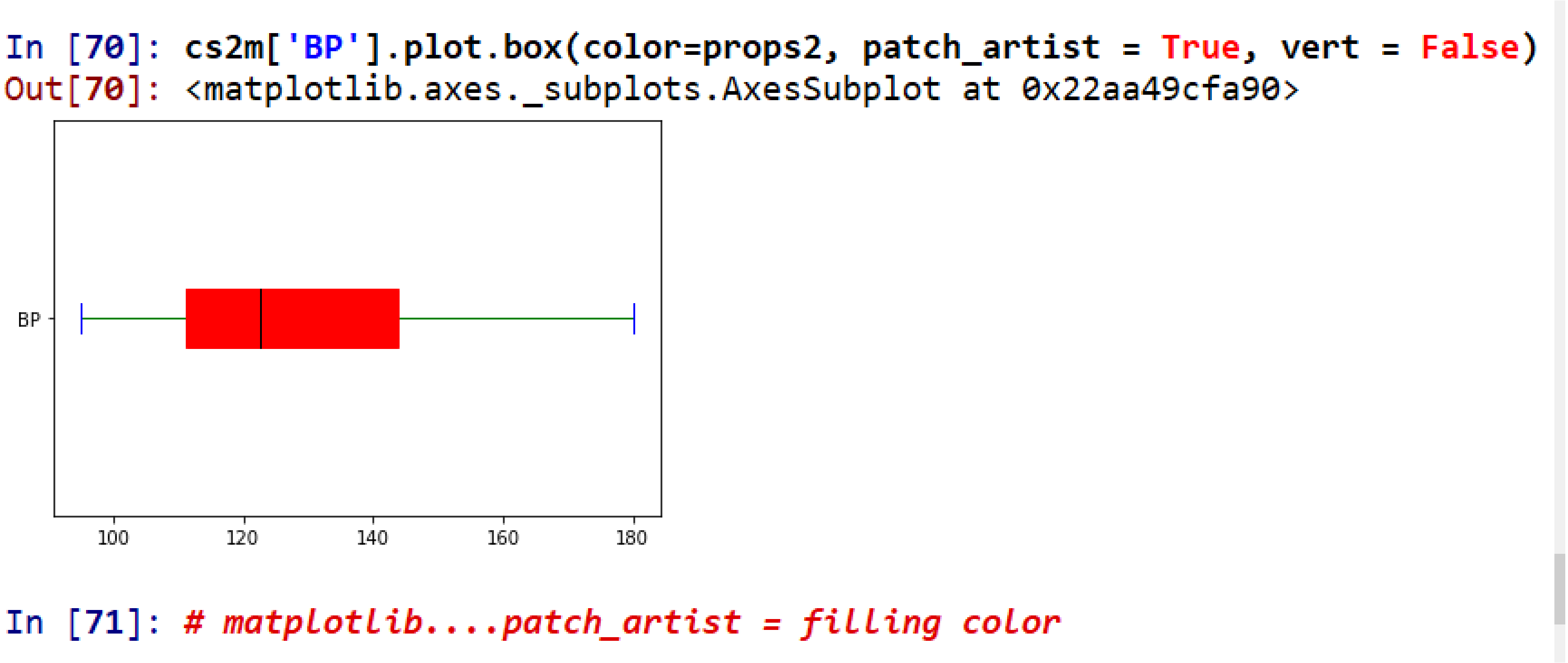
```
cs2m['BP'].plot.box(color=props2, patch_artist = True, vert = True)
```



```
In [69]: cs2m['BP'].plot.box(color=props2, patch_artist = True, vert = True)
Out[69]: <matplotlib.axes._subplots.AxesSubplot at 0x22aa4975278>
```

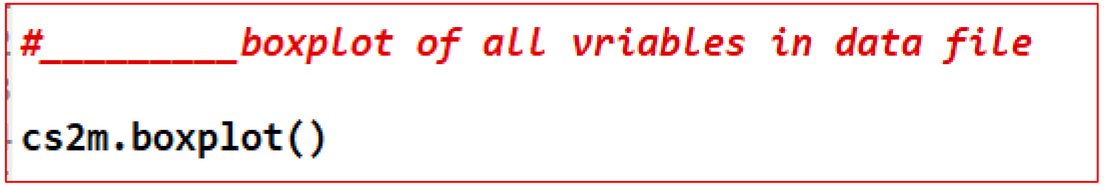
```
cs2m['BP'].plot.box(color=props2, patch_artist = True, vert = False)

# matplotlib....patch_artist = filling color
```

Box Plot



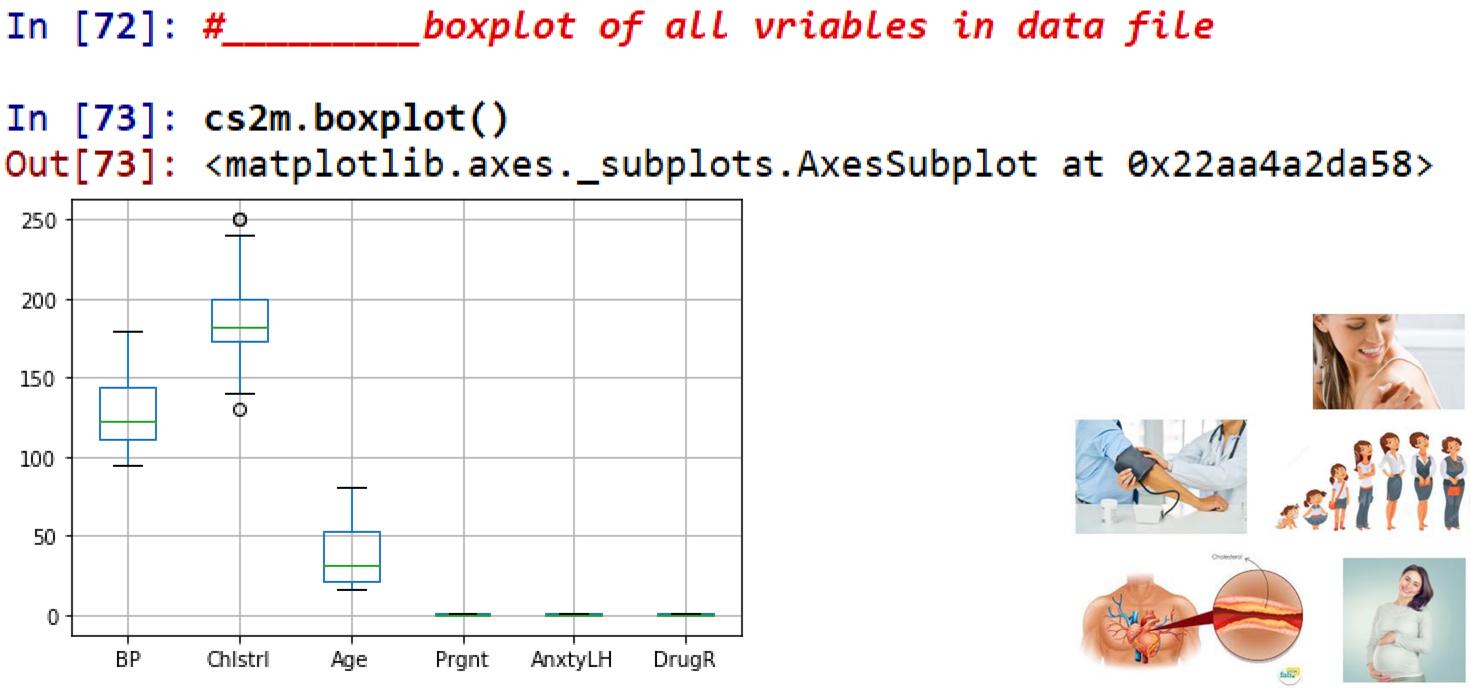
```
In [70]: cs2m['BP'].plot.box(color=props2, patch_artist = True, vert = False)
Out[70]: <matplotlib.axes._subplots.AxesSubplot at 0x22aa49cfa90>
```

```
In [71]: # matplotlib....patch_artist = filling color
```



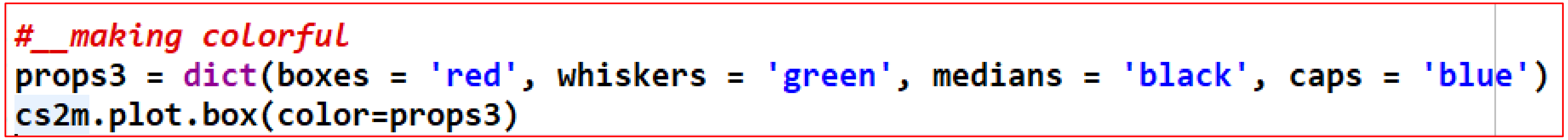
```
#_________boxplot of all vriables in data file

cs2m.boxplot()
```

Box Plot

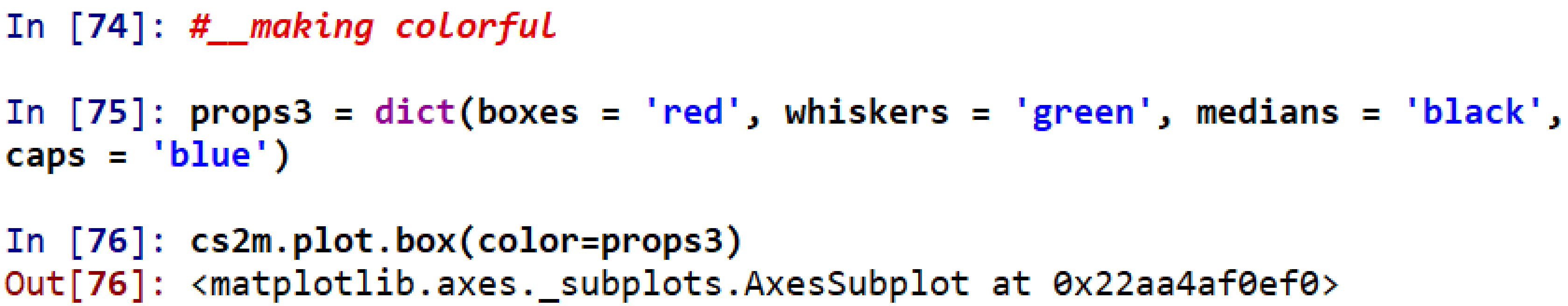


```
In [72]: #_________boxplot of all vriables in data file

In [73]: cs2m.boxplot()
Out[73]: <matplotlib.axes._subplots.AxesSubplot at 0x22aa4a2da58>
```

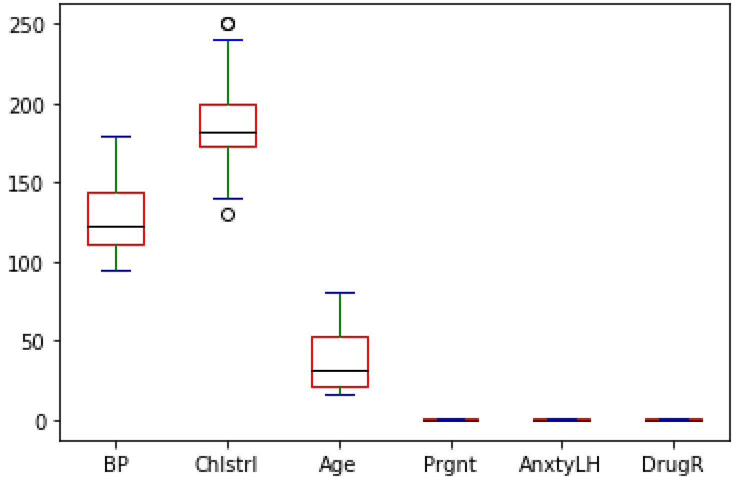
Box Plot



```
#__making colorful
props3 = dict(boxes = 'red', whiskers = 'green', medians = 'black', caps = 'blue')
cs2m.plot.box(color=props3)
```

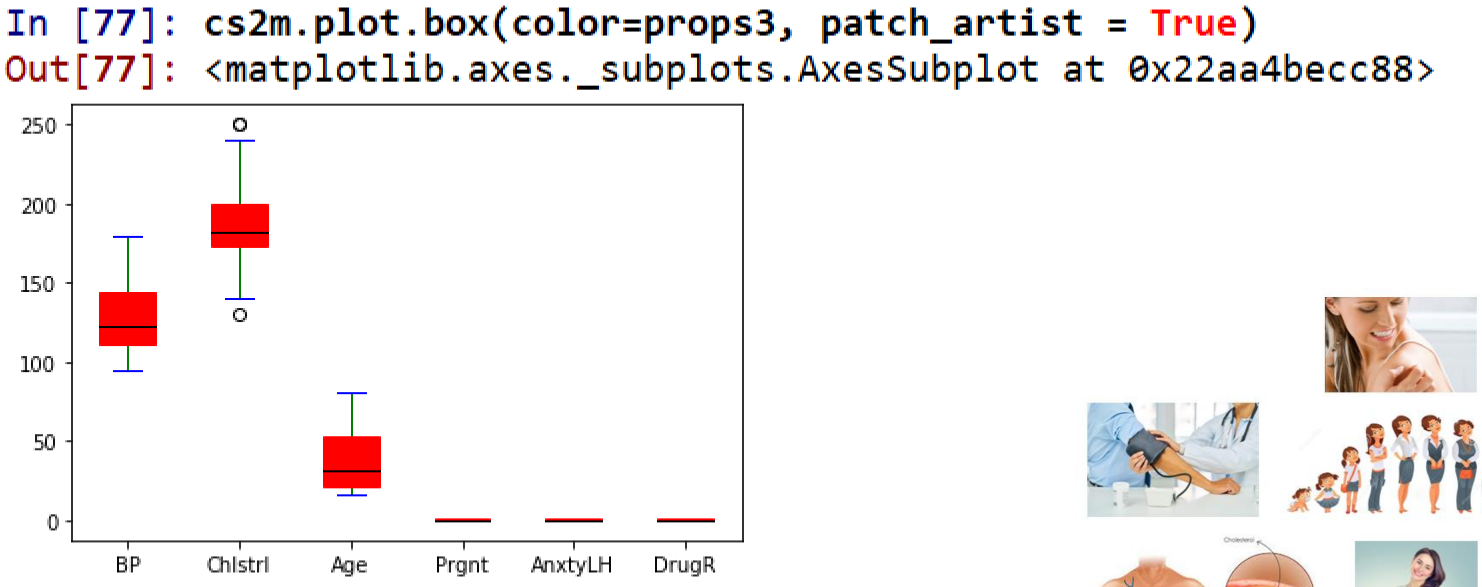


```
In [74]: #__making colorful

In [75]: props3 = dict(boxes = 'red', whiskers = 'green', medians = 'black',
caps = 'blue')

In [76]: cs2m.plot.box(color=props3)
Out[76]: <matplotlib.axes._subplots.AxesSubplot at 0x22aa4af0ef0>
```



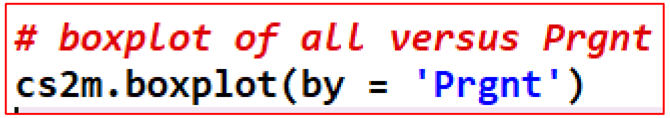
```
cs2m.plot.box(color=props3, patch_artist = True)
```

Box Plot

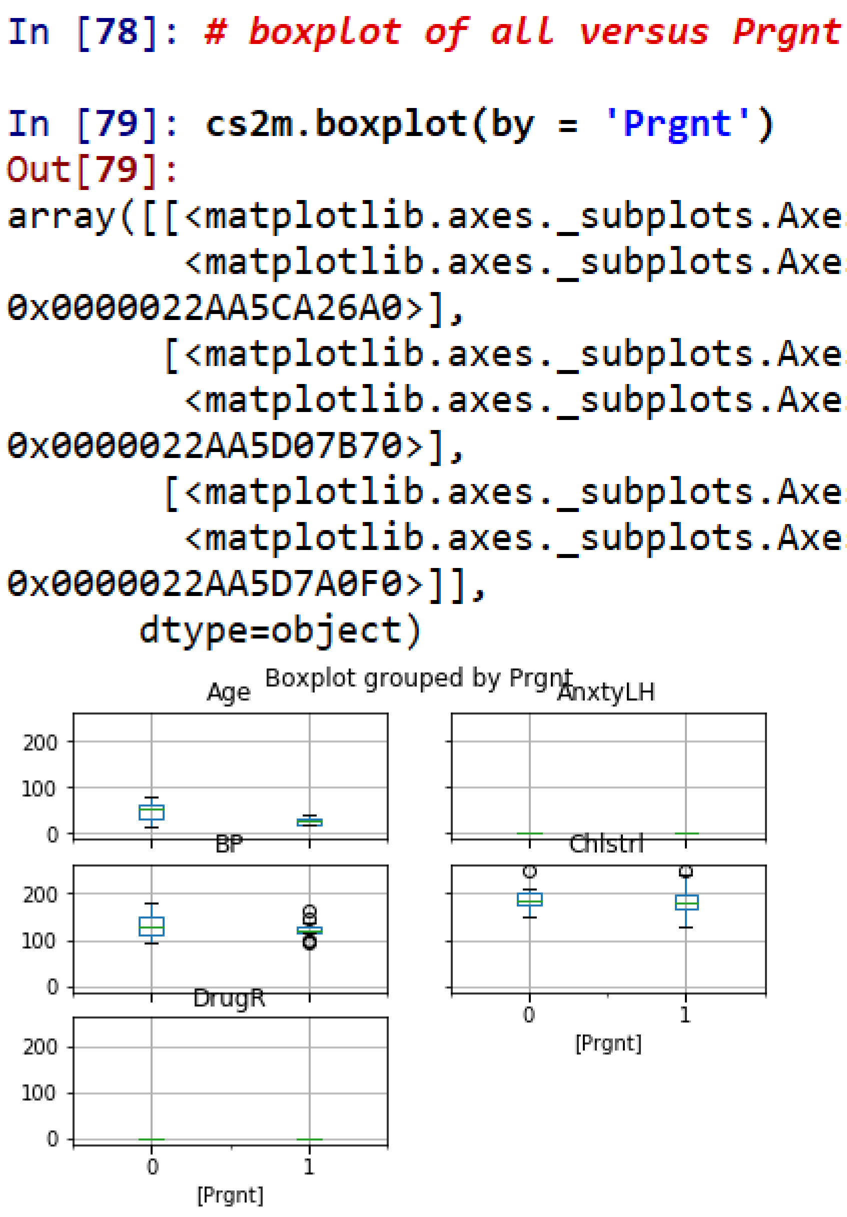


```
In [77]: cs2m.plot.box(color=props3, patch_artist = True)
Out[77]: <matplotlib.axes._subplots.AxesSubplot at 0x22aa4becc88>
```

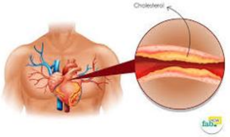
Box Plot



```
# boxplot of all versus Prgnt
cs2m.boxplot(by = 'Prgnt')
```



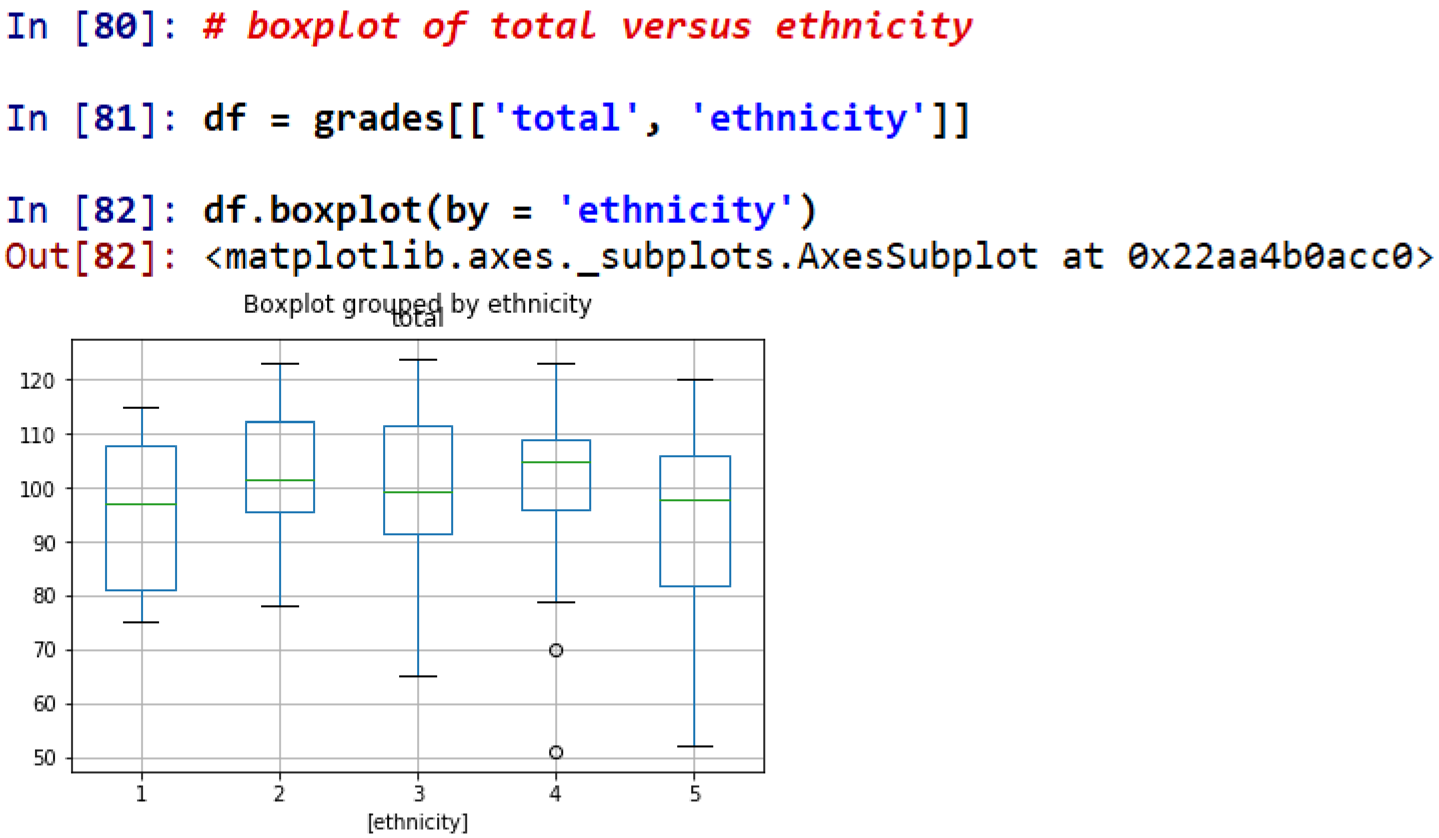
```
In [78]: # boxplot of all versus Prgnt

In [79]: cs2m.boxplot(by = 'Prgnt')
Out[79]:
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000022AA4C95780>,
        <matplotlib.axes._subplots.AxesSubplot object at
0x0000022AA5CA26A0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000022AA5CD38D0>,
        <matplotlib.axes._subplots.AxesSubplot object at
0x0000022AA5D07B70>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000022AA5D3CE10>,
        <matplotlib.axes._subplots.AxesSubplot object at
0x0000022AA5D7A0F0>]],
      dtype=object)
```



rstock.com • 1074559082

Box Plot

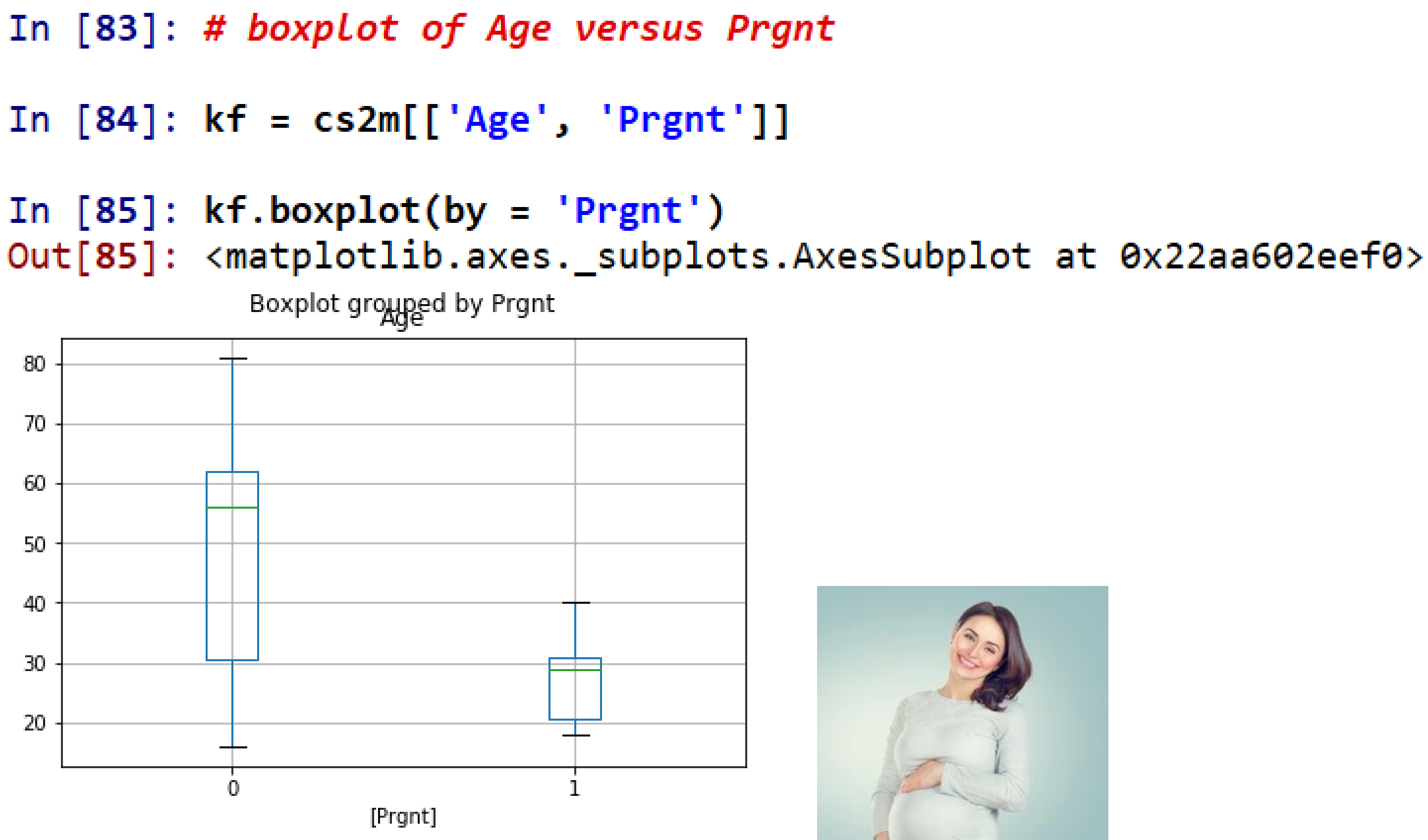
```
# boxplot of total versus ethnicity
df = grades[['total', 'ethnicity']]
df.boxplot(by = 'ethnicity')
```



```
In [80]: # boxplot of total versus ethnicity

In [81]: df = grades[['total', 'ethnicity']]

In [82]: df.boxplot(by = 'ethnicity')
Out[82]: <matplotlib.axes._subplots.AxesSubplot at 0x22aa4b0acc0>
```

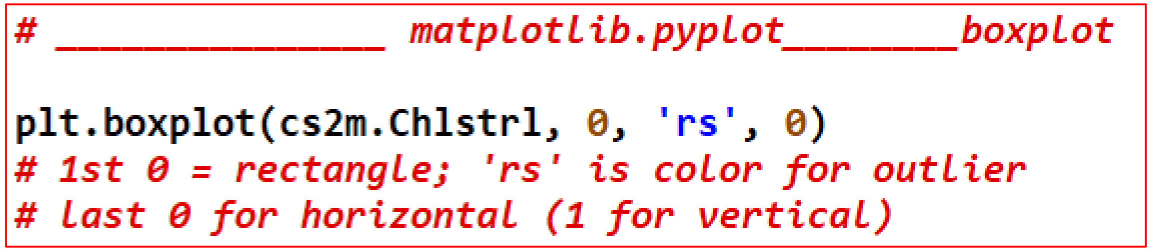
Box Plot

```
# boxplot of Age versus Prgnt
kf = cs2m[['Age', 'Prgnt']]
kf.boxplot(by = 'Prgnt')
```

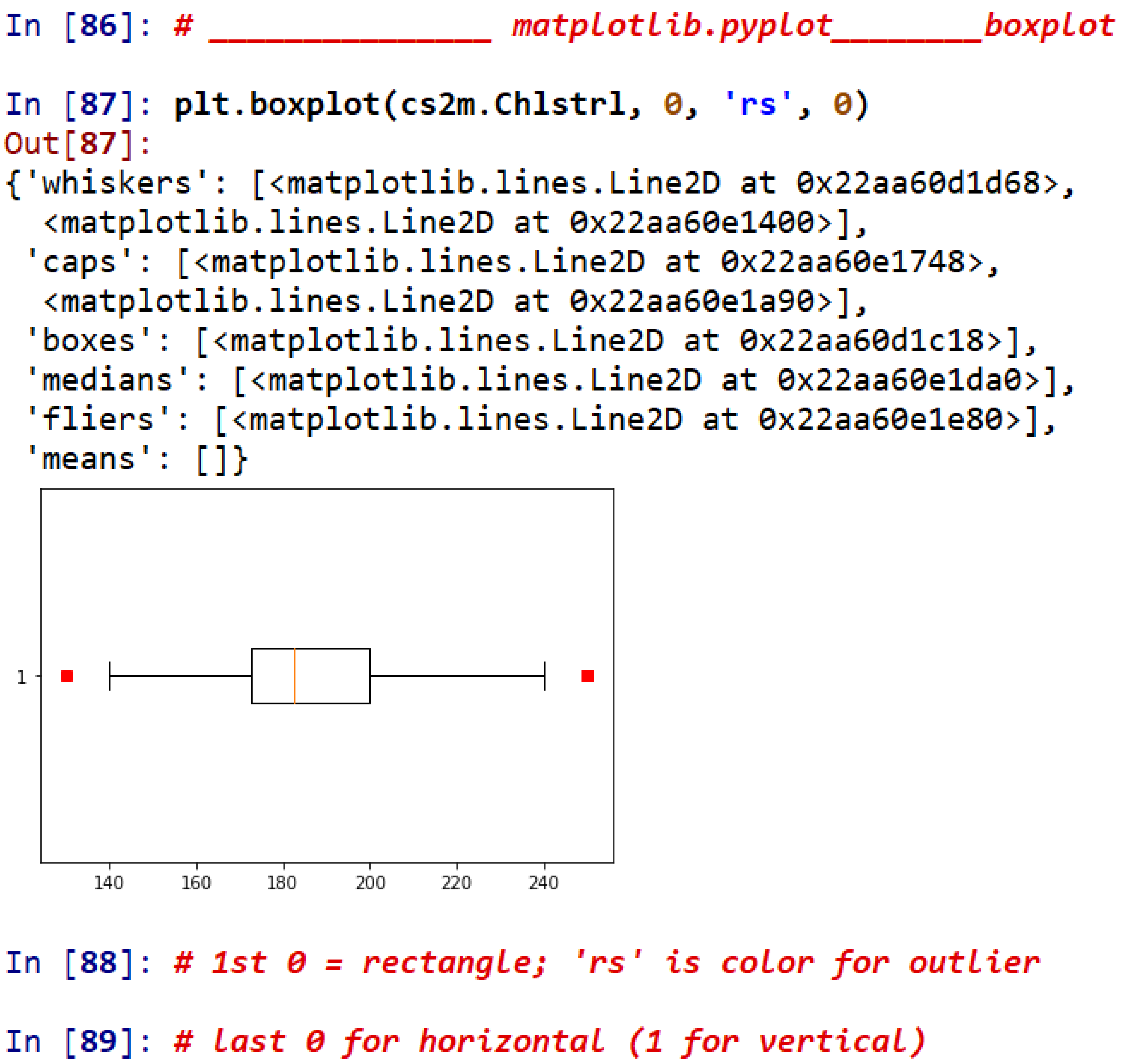


```
In [83]: # boxplot of Age versus Prgnt

In [84]: kf = cs2m[['Age', 'Prgnt']]

In [85]: kf.boxplot(by = 'Prgnt')
Out[85]: <matplotlib.axes._subplots.AxesSubplot at 0x22aa602eef0>
```

ɪrstock.com • 1074559082

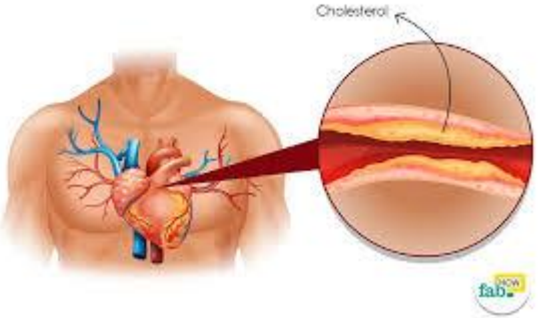


```
# _______________ matplotlib.pyplot________boxplot

plt.boxplot(cs2m.Chlstrl, 0, 'rs', 0)
# 1st 0 = rectangle; 'rs' is color for outlier
# last 0 for horizontal (1 for vertical)
```



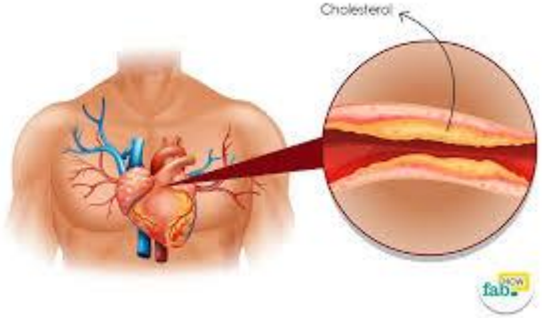
```
In [86]: # _______________ matplotlib.pyplot________boxplot

In [87]: plt.boxplot(cs2m.Chlstrl, 0, 'rs', 0)
Out[87]:
{'whiskers': [<matplotlib.lines.Line2D at 0x22aa60d1d68>,
  <matplotlib.lines.Line2D at 0x22aa60e1400>],
 'caps': [<matplotlib.lines.Line2D at 0x22aa60e1748>,
  <matplotlib.lines.Line2D at 0x22aa60e1a90>],
 'boxes': [<matplotlib.lines.Line2D at 0x22aa60d1c18>],
 'medians': [<matplotlib.lines.Line2D at 0x22aa60e1da0>],
 'fliers': [<matplotlib.lines.Line2D at 0x22aa60e1e80>],
 'means': []}
```

```
In [88]: # 1st 0 = rectangle; 'rs' is color for outlier

In [89]: # last 0 for horizontal (1 for vertical)
```

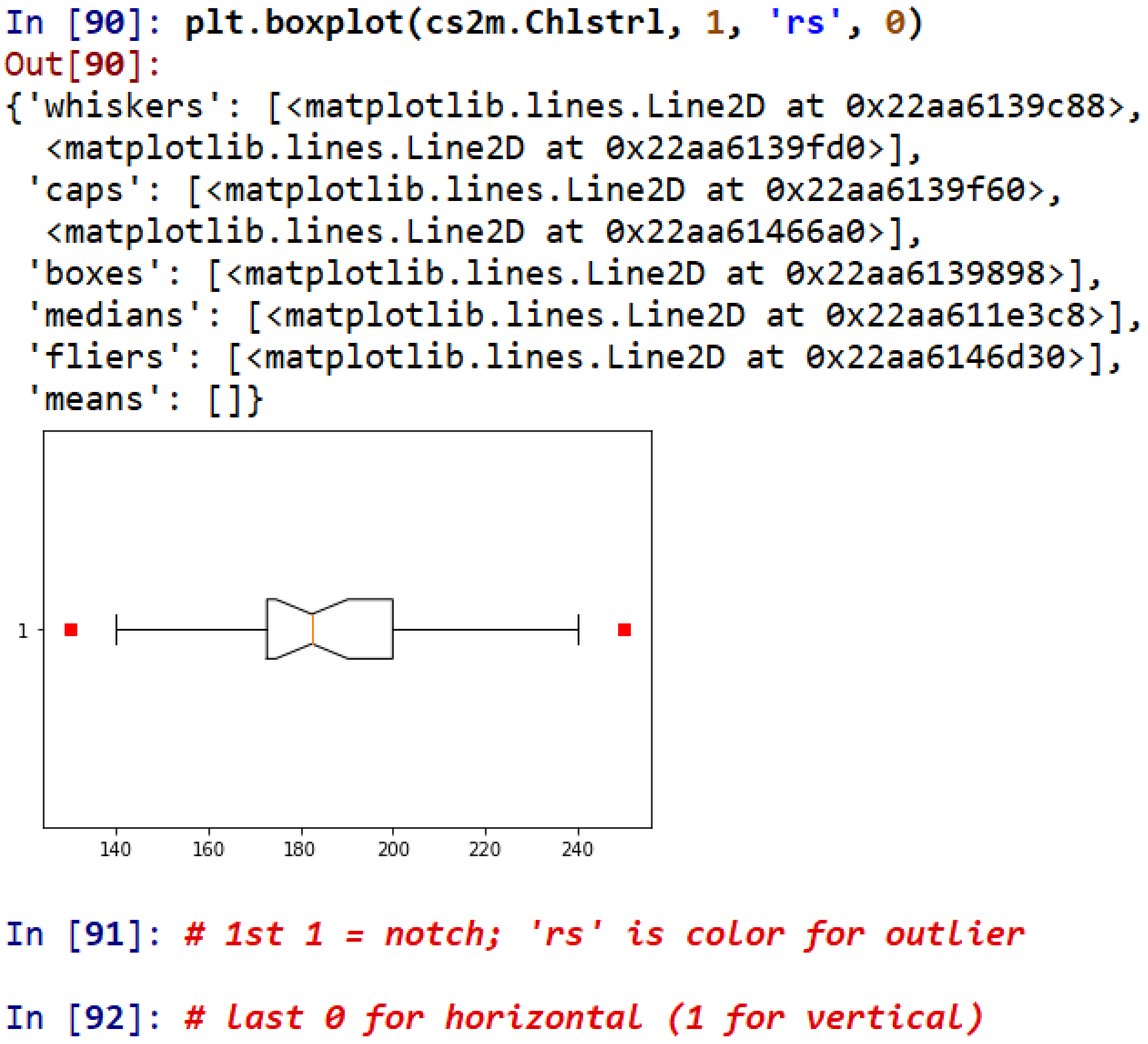


Box Plot

```
plt.boxplot(cs2m.Chlstrl, 1, 'rs', 0)
# 1st 1 = notch; 'rs' is color for outlier
# last 0 for horizontal (1 for vertical)
```



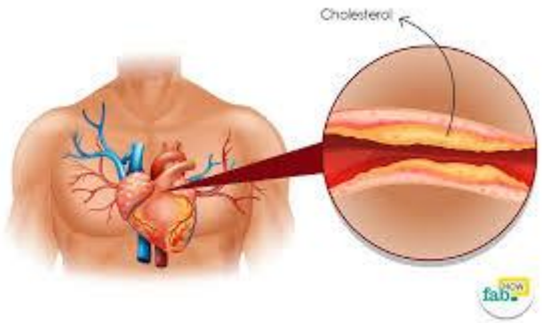
Box Plot



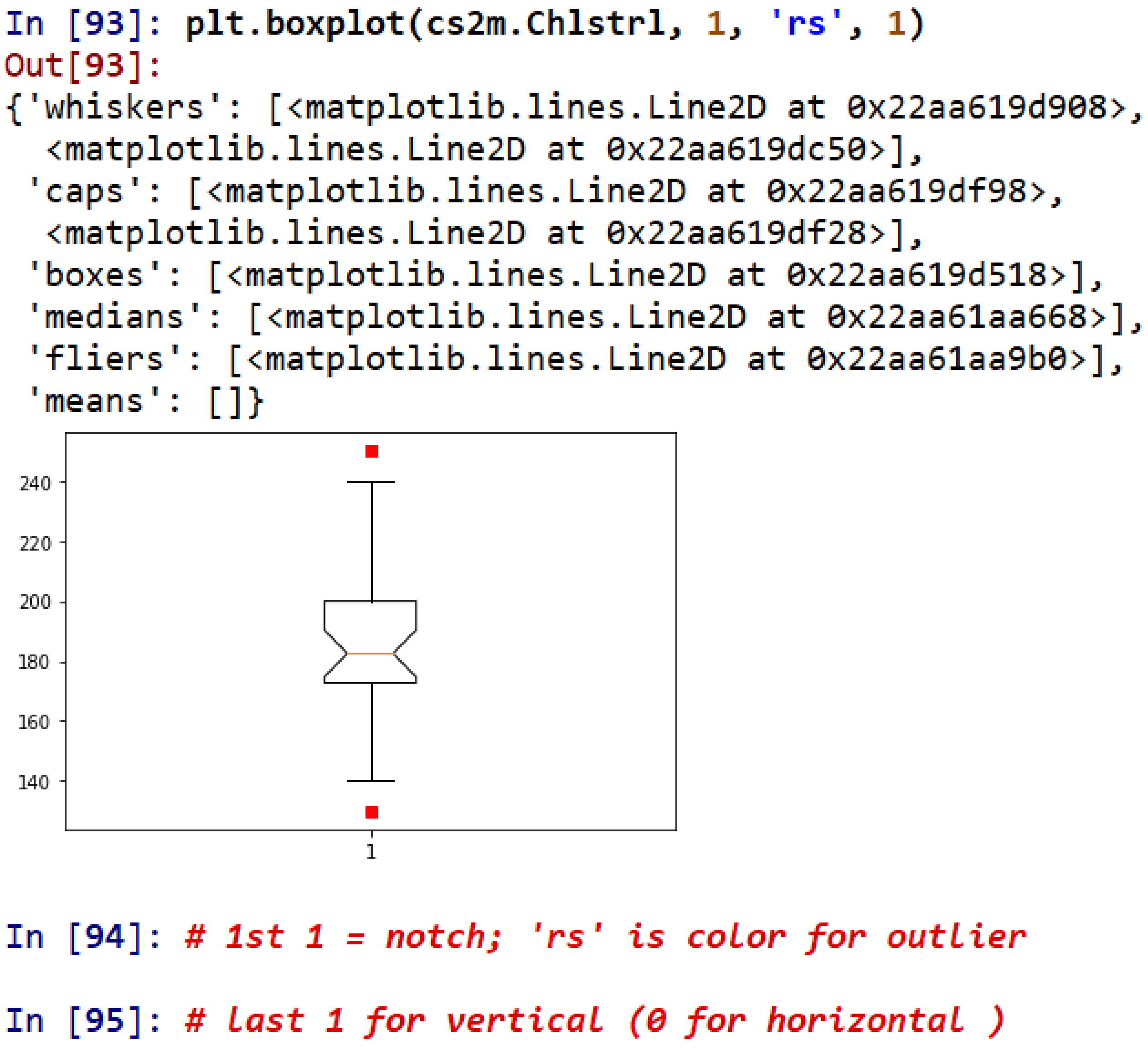
```
In [90]: plt.boxplot(cs2m.Chlstrl, 1, 'rs', 0)
Out[90]:
{'whiskers': [<matplotlib.lines.Line2D at 0x22aa6139c88>,
  <matplotlib.lines.Line2D at 0x22aa6139fd0>],
 'caps': [<matplotlib.lines.Line2D at 0x22aa6139f60>,
  <matplotlib.lines.Line2D at 0x22aa61466a0>],
 'boxes': [<matplotlib.lines.Line2D at 0x22aa6139898>],
 'medians': [<matplotlib.lines.Line2D at 0x22aa611e3c8>],
 'fliers': [<matplotlib.lines.Line2D at 0x22aa6146d30>],
 'means': []}
```

```
In [91]: # 1st 1 = notch; 'rs' is color for outlier

In [92]: # last 0 for horizontal (1 for vertical)
```

```
plt.boxplot(cs2m.Chlstrl, 1, 'rs', 1)
# 1st 1 = notch; 'rs' is color for outlier
# last 1 for vertical (0 for horizontal )
```



Box Plot

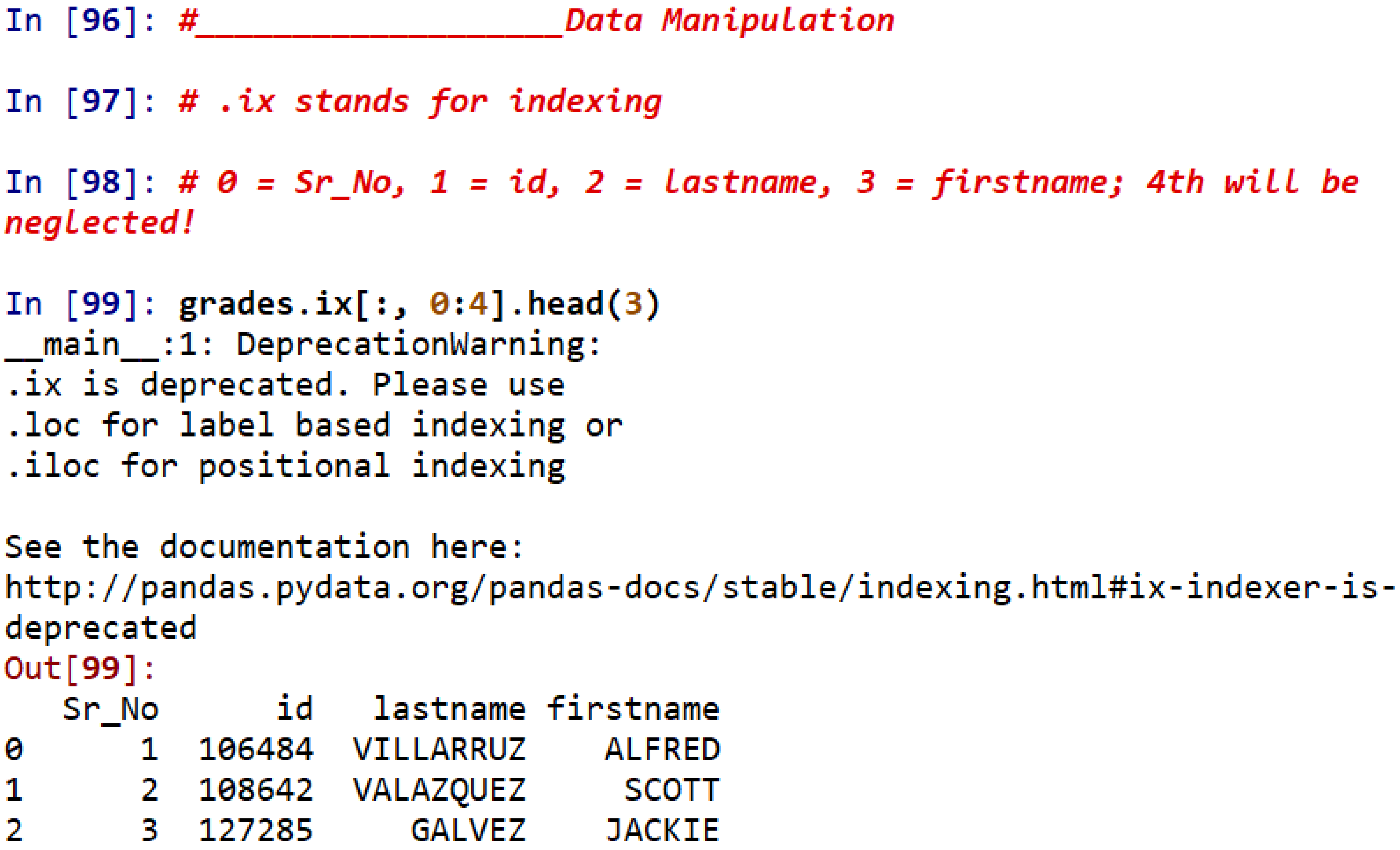


```
In [93]: plt.boxplot(cs2m.Chlstrl, 1, 'rs', 1)
Out[93]:
{'whiskers': [<matplotlib.lines.Line2D at 0x22aa619d908>,
  <matplotlib.lines.Line2D at 0x22aa619dc50>],
 'caps': [<matplotlib.lines.Line2D at 0x22aa619df98>,
  <matplotlib.lines.Line2D at 0x22aa619df28>],
 'boxes': [<matplotlib.lines.Line2D at 0x22aa619d518>],
 'medians': [<matplotlib.lines.Line2D at 0x22aa61aa668>],
 'fliers': [<matplotlib.lines.Line2D at 0x22aa61aa9b0>],
 'means': []}
```

```
In [94]: # 1st 1 = notch; 'rs' is color for outlier

In [95]: # last 1 for vertical (0 for horizontal )
```

```
#___________________Data Manipulation

# .ix stands for indexing
# 0 = Sr_No, 1 = id, 2 = lastname, 3 = firstname; 4th will be neglected!
grades.ix[:, 0:4].head(3)
```

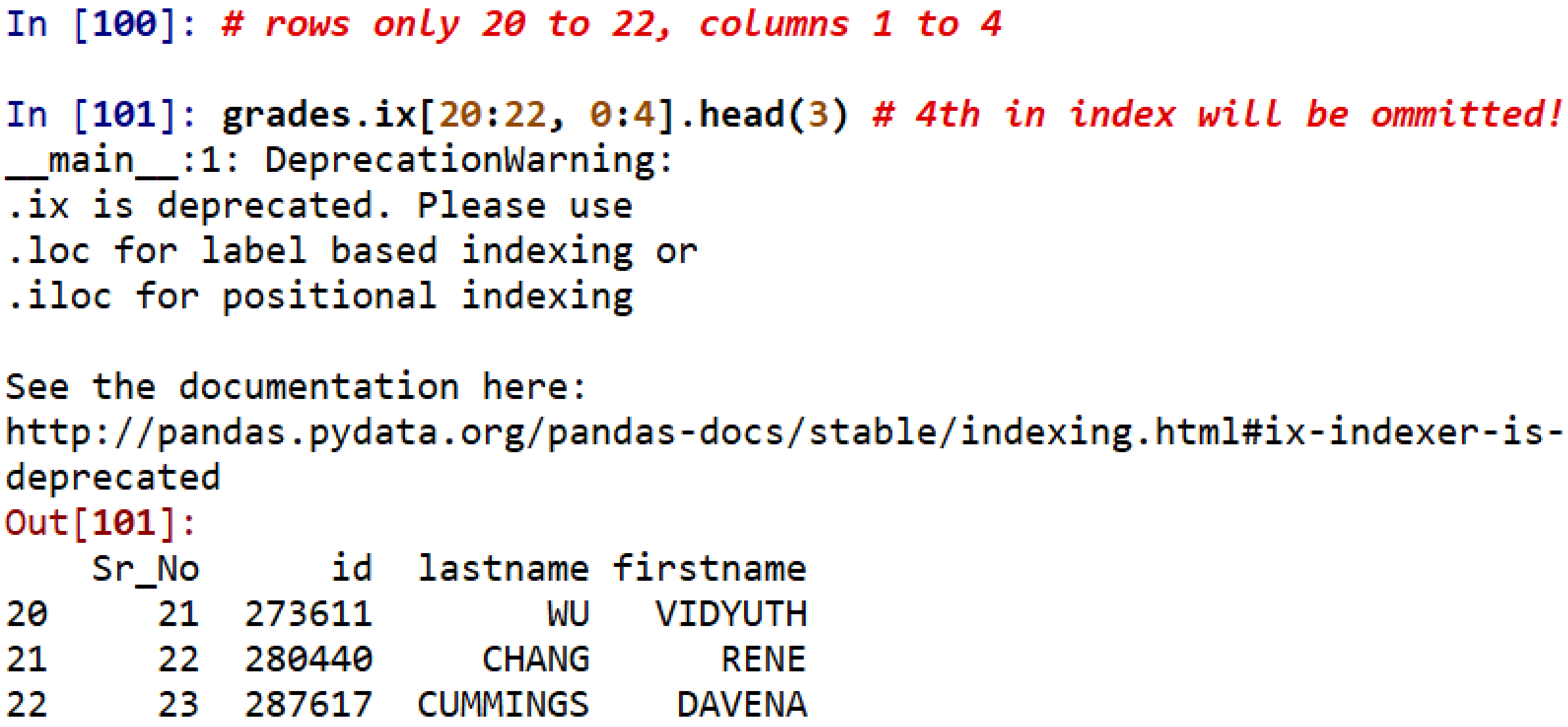
Selection



```
In [96]: #___________________Data Manipulation

In [97]: # .ix stands for indexing

In [98]: # 0 = Sr_No, 1 = id, 2 = lastname, 3 = firstname; 4th will be
neglected!

In [99]: grades.ix[:, 0:4].head(3)
__main__:1: DeprecationWarning:
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#ix-indexer-is-
deprecated
Out[99]:
   Sr_No      id   lastname firstname
0      1  106484  VILLARRUZ    ALFRED
1      2  108642  VALAZQUEZ     SCOTT
2      3  127285     GALVEZ    JACKIE
```

```
# rows only 20 to 22, columns 1 to 4
grades.ix[20:22, 0:4].head(3) # 4th in index will be ommitted!
```

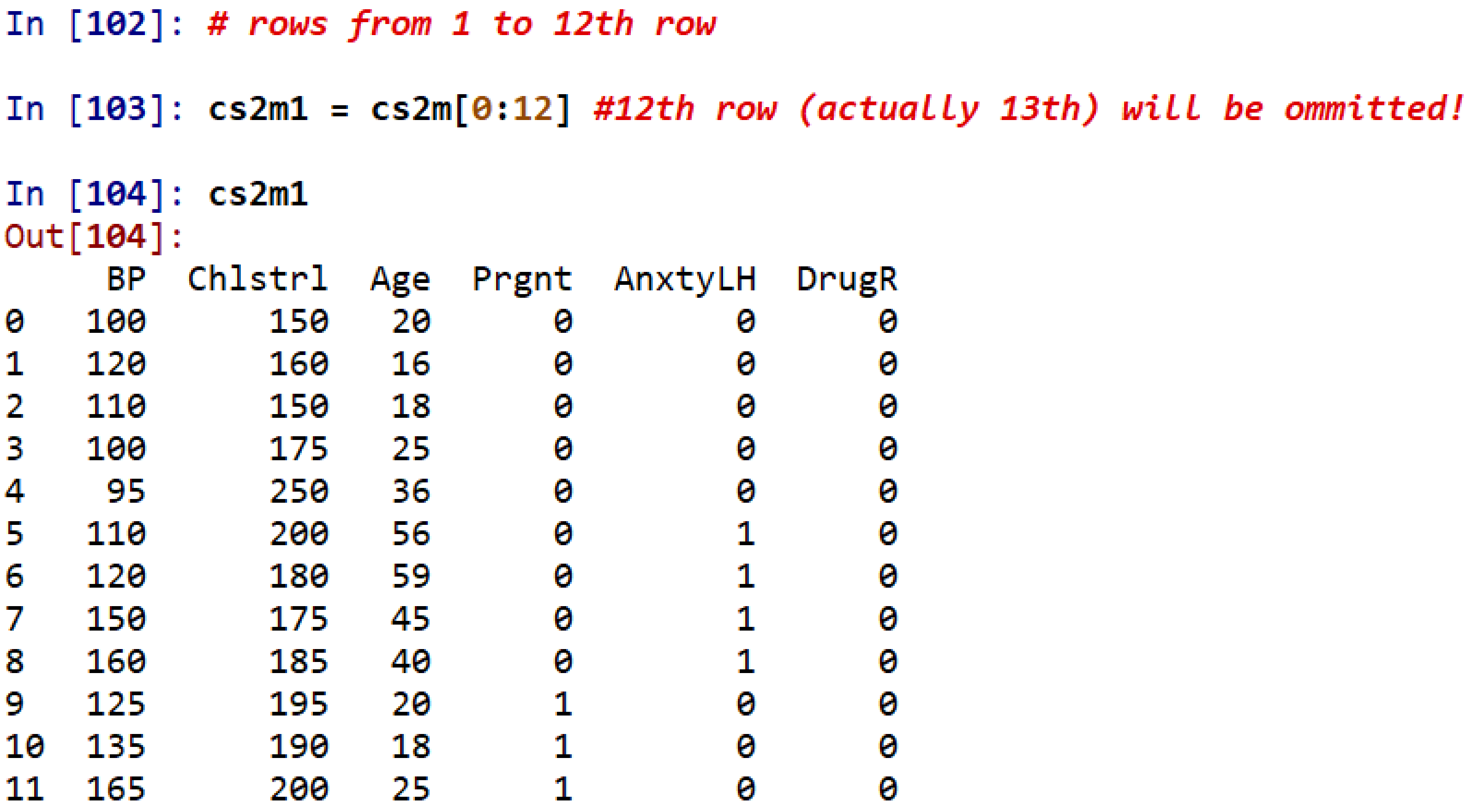
Selection



```
In [100]: # rows only 20 to 22, columns 1 to 4

In [101]: grades.ix[20:22, 0:4].head(3) # 4th in index will be ommitted!
__main__:1: DeprecationWarning:
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#ix-indexer-is-
deprecated
Out[101]:
    Sr_No      id  lastname firstname
20     21  273611        WU   VIDYUTH
21     22  280440     CHANG      RENE
22     23  287617  CUMMINGS    DAVENA
```

```
# rows from 1 to 12th row
cs2m1 = cs2m[0:12] #12th row (actually 13th) will be ommitted!
cs2m1
cs2m1.head()
```

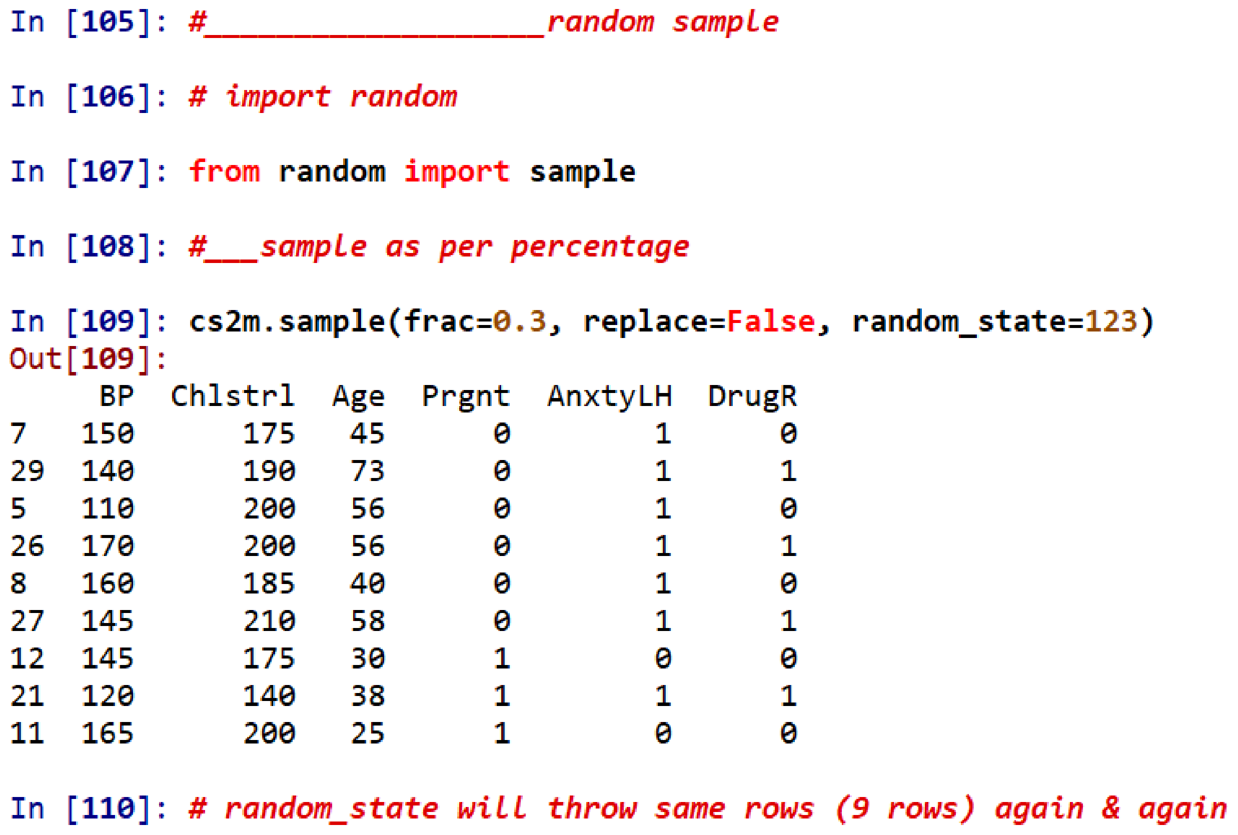
Selection



```
In [102]: # rows from 1 to 12th row

In [103]: cs2m1 = cs2m[0:12] #12th row (actually 13th) will be ommitted!

In [104]: cs2m1
Out[104]:
     BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR
0   100      150   20      0        0      0
1   120      160   16      0        0      0
2   110      150   18      0        0      0
3   100      175   25      0        0      0
4    95      250   36      0        0      0
5   110      200   56      0        1      0
6   120      180   59      0        1      0
7   150      175   45      0        1      0
8   160      185   40      0        1      0
9   125      195   20      1        0      0
10  135      190   18      1        0      0
11  165      200   25      1        0      0
```

```
#___________________random sample

# import random
from random import sample

#___sample as per percentage
cs2m.sample(frac=0.3, replace=False, random_state=123)
# random_state will throw same rows (9 rows) again & again
```

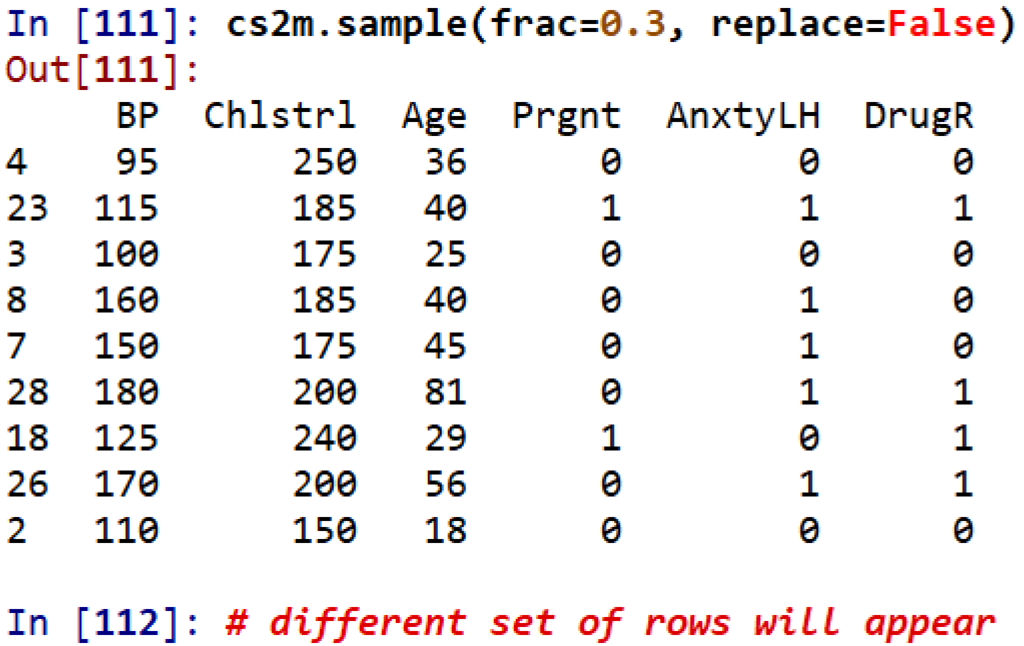
Selection



```
In [105]: #_________________random sample

In [106]: # import random

In [107]: from random import sample

In [108]: #___sample as per percentage

In [109]: cs2m.sample(frac=0.3, replace=False, random_state=123)
Out[109]:
     BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR
7   150      175   45      0        1      0
29  140      190   73      0        1      1
5   110      200   56      0        1      0
26  170      200   56      0        1      1
8   160      185   40      0        1      0
27  145      210   58      0        1      1
12  145      175   30      1        0      0
21  120      140   38      1        1      1
11  165      200   25      1        0      0

In [110]: # random_state will throw same rows (9 rows) again & again
```

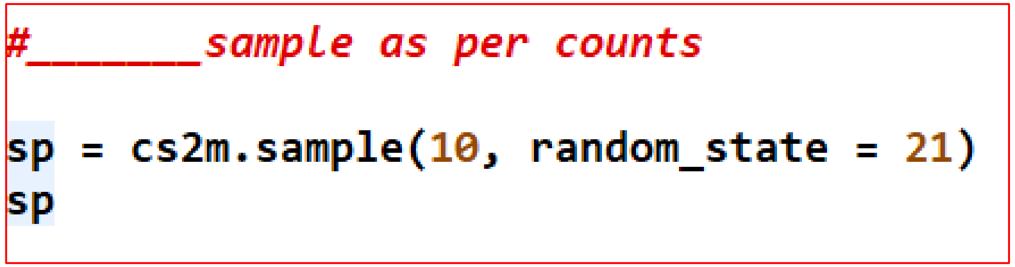
```
cs2m.sample(frac=0.3, replace=False)
# different set of rows will appear
```

Selection



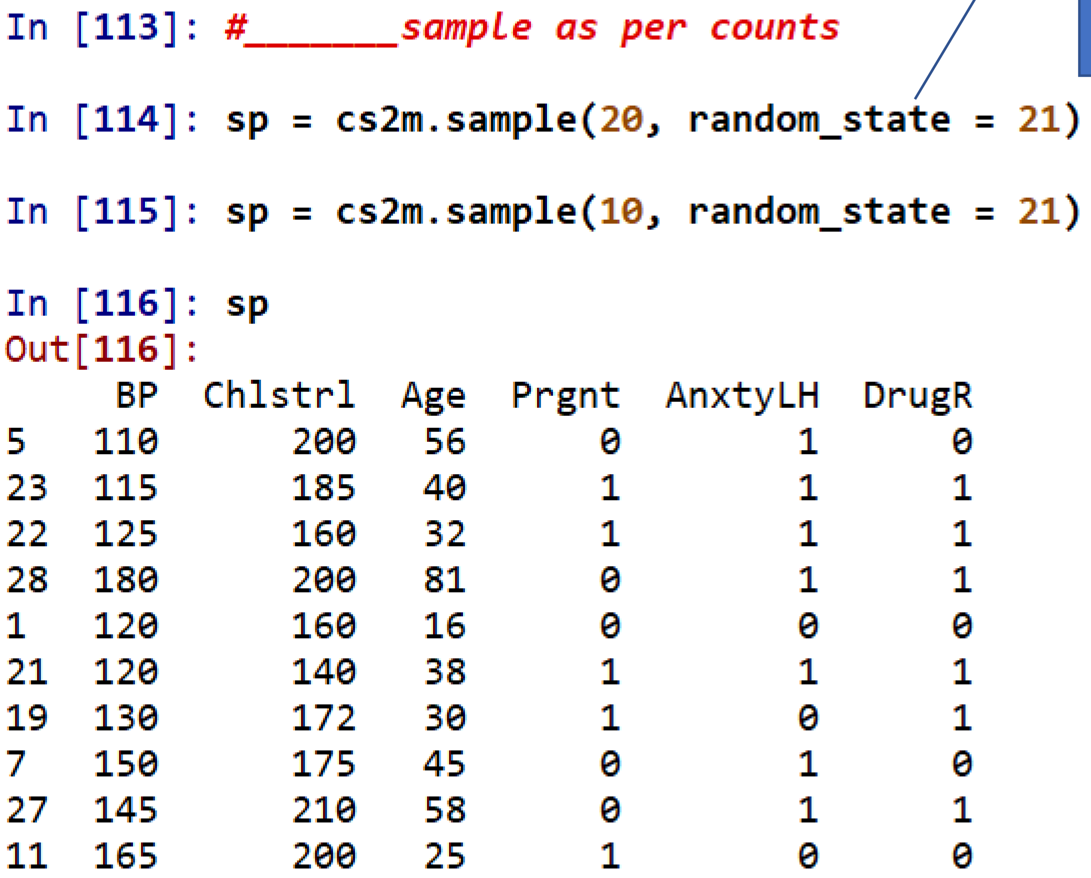
```
In [111]: cs2m.sample(frac=0.3, replace=False)
Out[111]:
     BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR
4    95      250   36      0        0      0
23  115      185   40      1        1      1
3   100      175   25      0        0      0
8   160      185   40      0        1      0
7   150      175   45      0        1      0
28  180      200   81      0        1      1
18  125      240   29      1        0      1
26  170      200   56      0        1      1
2   110      150   18      0        0      0

In [112]: # different set of rows will appear
```

Selection

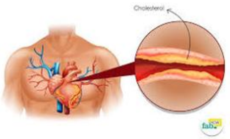


```
#_______sample as per counts

sp = cs2m.sample(10, random_state = 21)
sp
```

Ignore this line

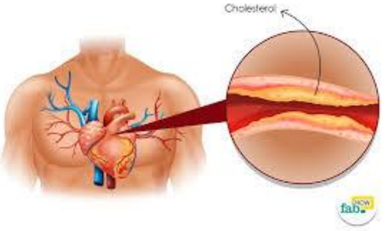


```
In [113]: #_______sample as per counts

In [114]: sp = cs2m.sample(20, random_state = 21)

In [115]: sp = cs2m.sample(10, random_state = 21)

In [116]: sp
Out[116]:
     BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR
5   110      200   56      0        1      0
23  115      185   40      1        1      1
22  125      160   32      1        1      1
28  180      200   81      0        1      1
1   120      160   16      0        0      0
21  120      140   38      1        1      1
19  130      172   30      1        0      1
7   150      175   45      0        1      0
27  145      210   58      0        1      1
11  165      200   25      1        0      0
```



Selection

```
#______selecting choiced variables, all rows

# all rows and columns 1,3,5
# 0 is sr_no, will be ignored
cs2m.ix[:, (1, 3, 5)].head(3)
```

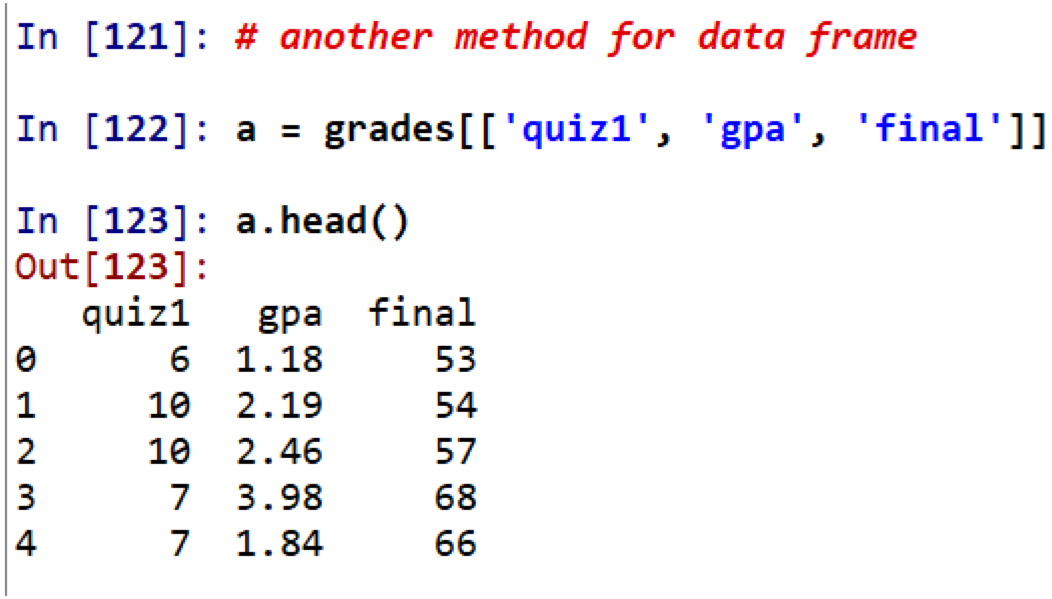


ɪrstock.com • 1074559082

```
In [117]: #______selecting choiced variables, all rows

In [118]: # all rows and columns 1,3,5

In [119]: # 0 is sr_no, will be ignored

In [120]: cs2m.ix[:, (1, 3, 5)].head(3)
__main__:1: DeprecationWarning:
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#ix-indexer-is-
deprecated
Out[120]:
   Chlstrl  Prgnt  DrugR
0      150      0      0
1      160      0      0
2      150      0      0
```

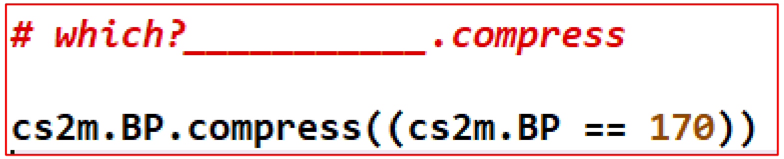
Selection

```
# another method for data frame

a = grades[['quiz1', 'gpa', 'final']]
a.head()
```

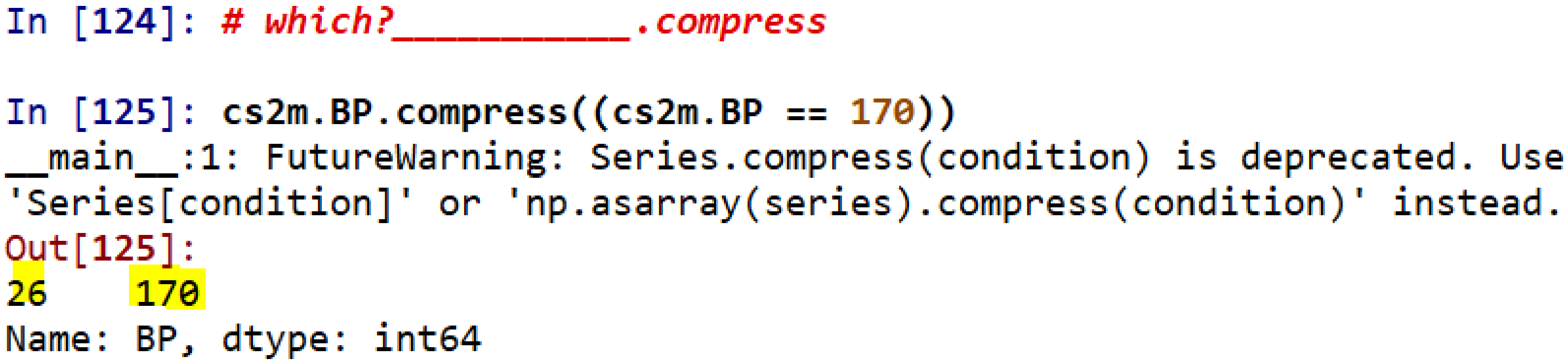


```
In [121]: # another method for data frame

In [122]: a = grades[['quiz1', 'gpa', 'final']]

In [123]: a.head()
Out[123]:
   quiz1   gpa  final
0      6  1.18     53
1     10  2.19     54
2     10  2.46     57
3      7  3.98     68
4      7  1.84     66
```

Selection



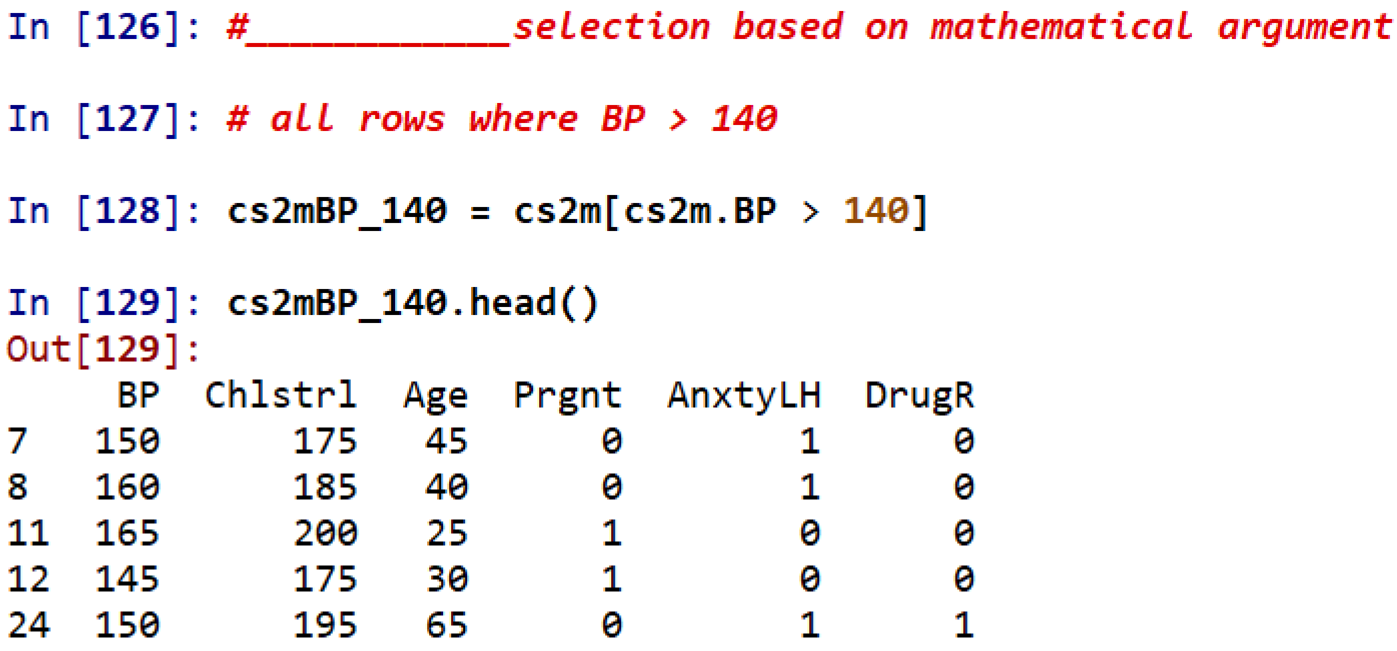
```
# which?__________.compress

cs2m.BP.compress((cs2m.BP == 170))
```



```
In [124]: # which?__________.compress

In [125]: cs2m.BP.compress((cs2m.BP == 170))
__main__:1: FutureWarning: Series.compress(condition) is deprecated. Use
'Series[condition]' or 'np.asarray(series).compress(condition)' instead.
Out[125]:
26    170
Name: BP, dtype: int64
```

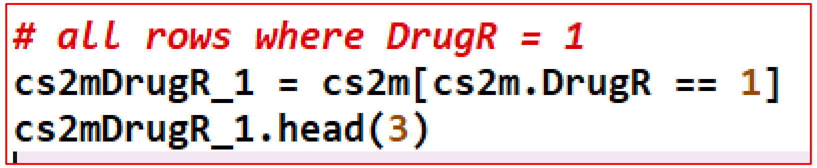
Selection

```
#____________selection based on mathematical argument

# all rows where BP > 140
cs2mBP_140 = cs2m[cs2m.BP > 140]
cs2mBP_140.head()
```

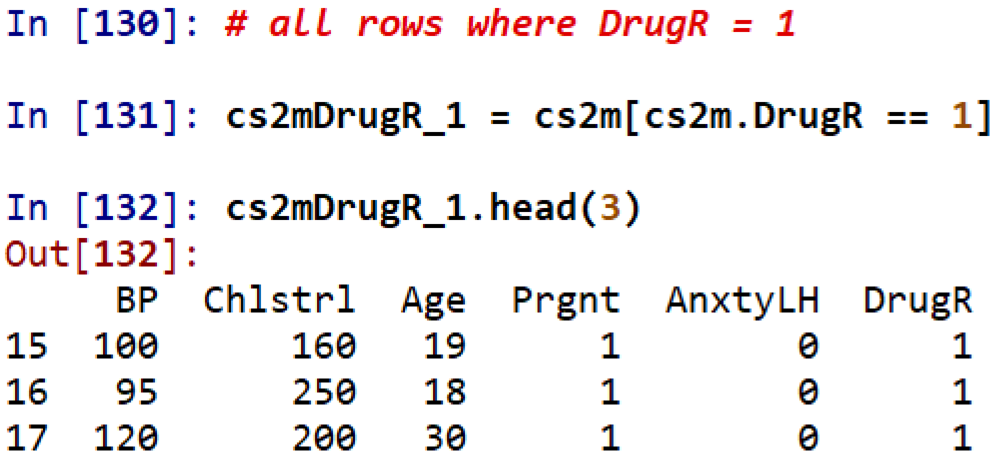


```
In [126]: #____________selection based on mathematical argument

In [127]: # all rows where BP > 140

In [128]: cs2mBP_140 = cs2m[cs2m.BP > 140]

In [129]: cs2mBP_140.head()
Out[129]:
     BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR
7   150      175   45      0        1      0
8   160      185   40      0        1      0
11  165      200   25      1        0      0
12  145      175   30      1        0      0
24  150      195   65      0        1      1
```

Selection

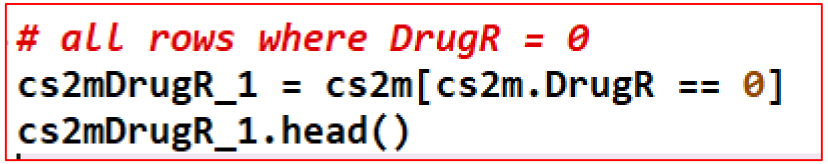


```
# all rows where DrugR = 1
cs2mDrugR_1 = cs2m[cs2m.DrugR == 1]
cs2mDrugR_1.head(3)
```

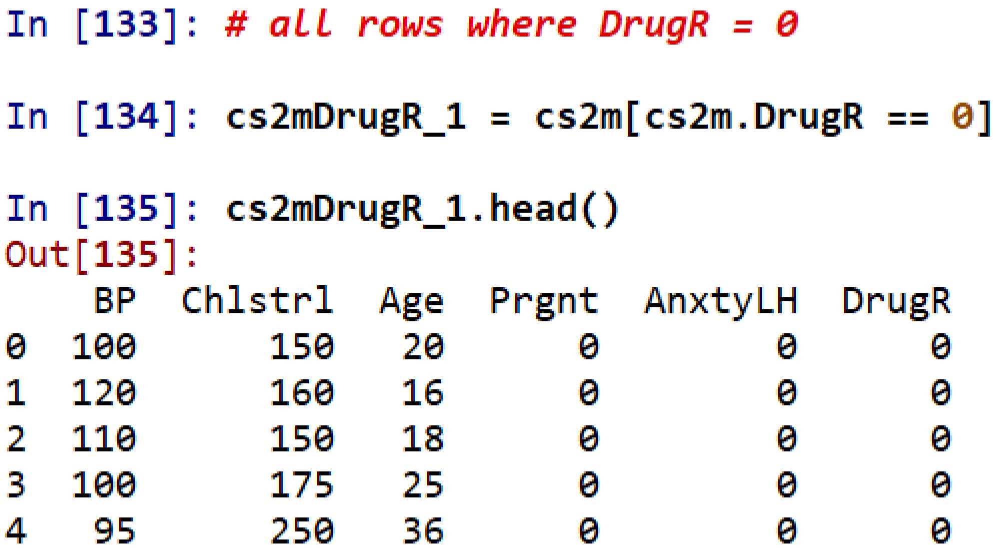


```
In [130]: # all rows where DrugR = 1

In [131]: cs2mDrugR_1 = cs2m[cs2m.DrugR == 1]

In [132]: cs2mDrugR_1.head(3)
Out[132]:
     BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR
15  100      160   19      1        0      1
16   95      250   18      1        0      1
17  120      200   30      1        0      1
```

Selection



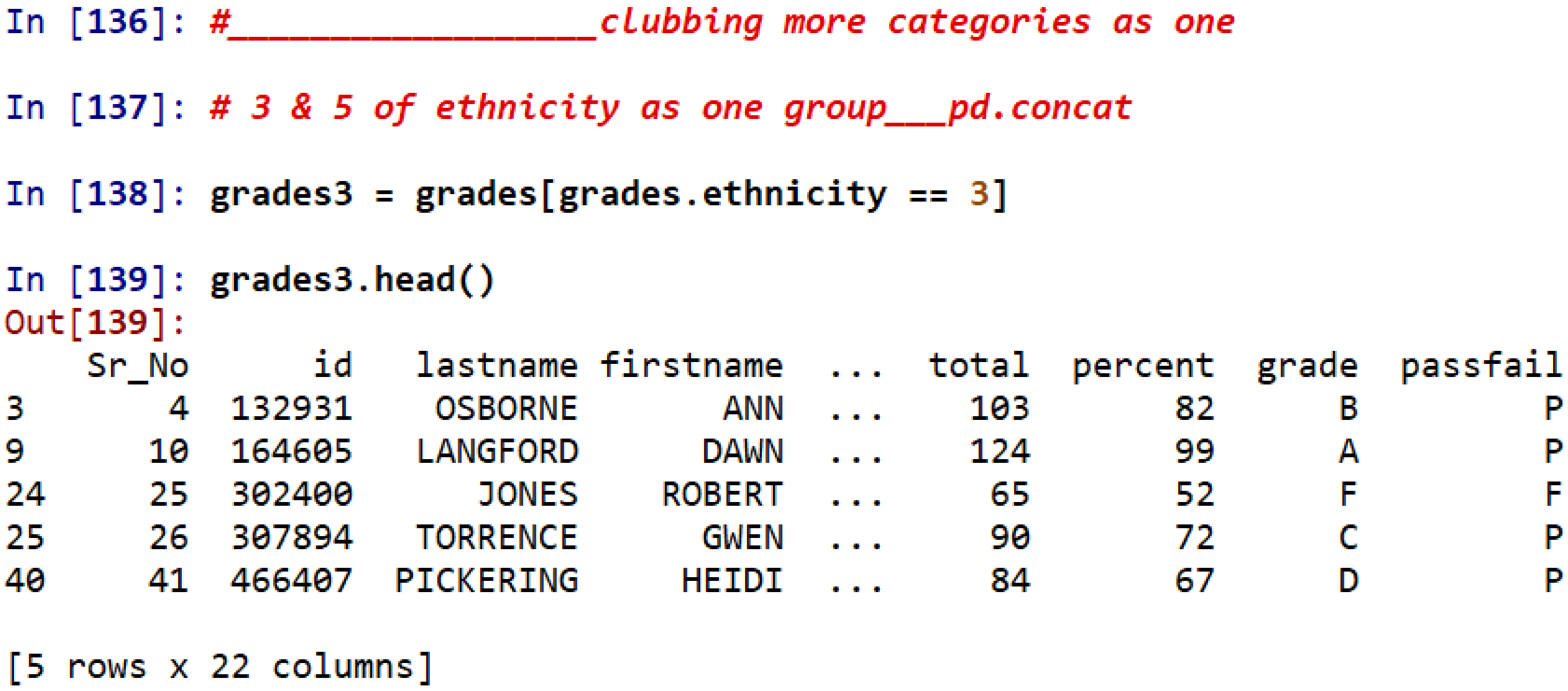
```
# all rows where DrugR = 0
cs2mDrugR_1 = cs2m[cs2m.DrugR == 0]
cs2mDrugR_1.head()
```



```
In [133]: # all rows where DrugR = 0

In [134]: cs2mDrugR_1 = cs2m[cs2m.DrugR == 0]

In [135]: cs2mDrugR_1.head()
Out[135]:
    BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR
0  100      150   20      0        0      0
1  120      160   16      0        0      0
2  110      150   18      0        0      0
3  100      175   25      0        0      0
4   95      250   36      0        0      0
```

```
#__________________clubbing more categories as one

# 3 & 5 of ethnicity as one group___pd.concat
grades3 = grades[grades.ethnicity == 3]
grades3.head()
```

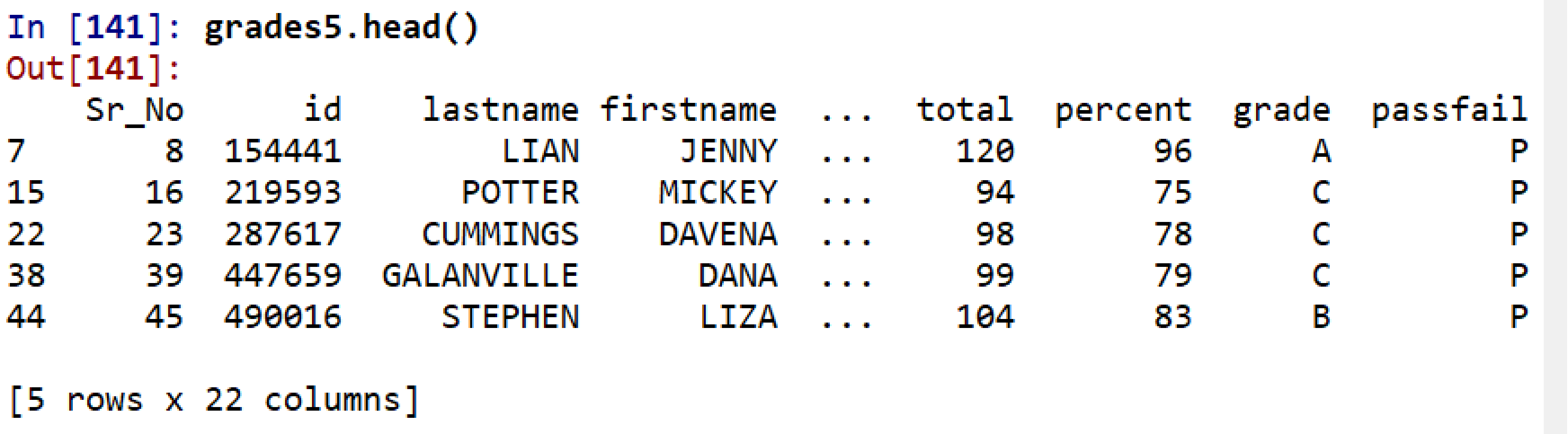
Selection & Grouping



```
In [136]: #__________________clubbing more categories as one

In [137]: # 3 & 5 of ethnicity as one group___pd.concat

In [138]: grades3 = grades[grades.ethnicity == 3]

In [139]: grades3.head()
Out[139]:
    Sr_No      id   lastname firstname  ...  total  percent  grade  passfail
3       4  132931    OSBORNE       ANN  ...    103       82      B         P
9      10  164605   LANGFORD      DAWN  ...    124       99      A         P
24     25  302400      JONES    ROBERT  ...     65       52      F         F
25     26  307894   TORRENCE      GWEN  ...     90       72      C         P
40     41  466407  PICKERING     HEIDI  ...     84       67      D         P

[5 rows x 22 columns]
```

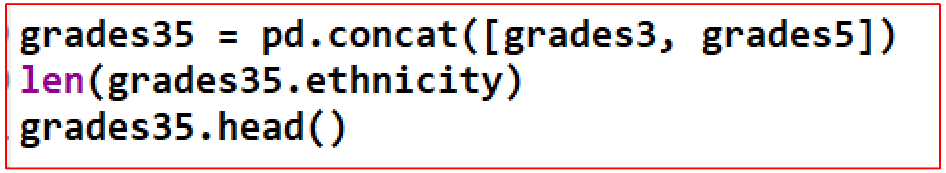
Selection & Grouping

```
grades5 = grades[grades.ethnicity == 5]
grades5.head()
```

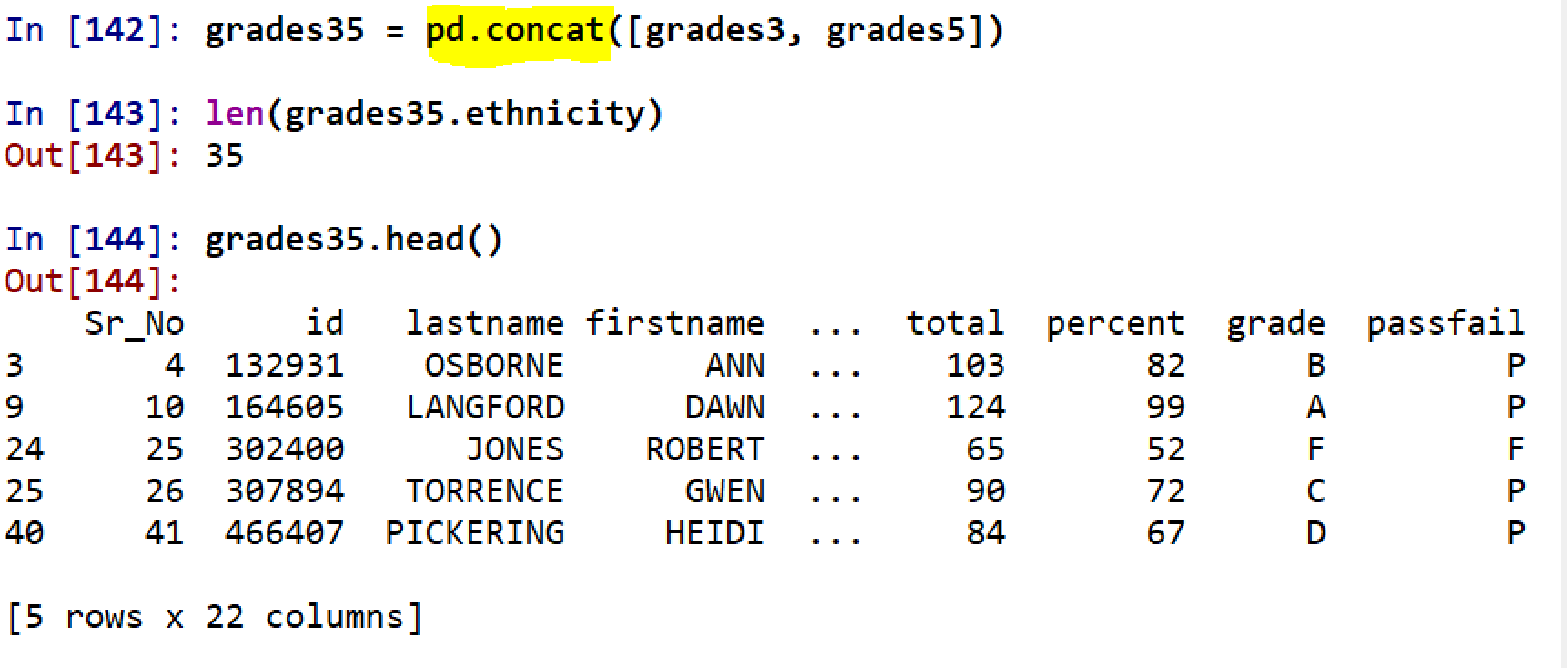


```
In [141]: grades5.head()
Out[141]:
    Sr_No      id    lastname firstname  ...  total  percent  grade  passfail
7       8  154441        LIAN     JENNY  ...    120       96      A         P
15     16  219593      POTTER    MICKEY  ...     94       75      C         P
22     23  287617    CUMMINGS    DAVENA  ...     98       78      C         P
38     39  447659  GALANVILLE      DANA  ...     99       79      C         P
44     45  490016     STEPHEN      LIZA  ...    104       83      B         P

[5 rows x 22 columns]
```

Selection & Grouping



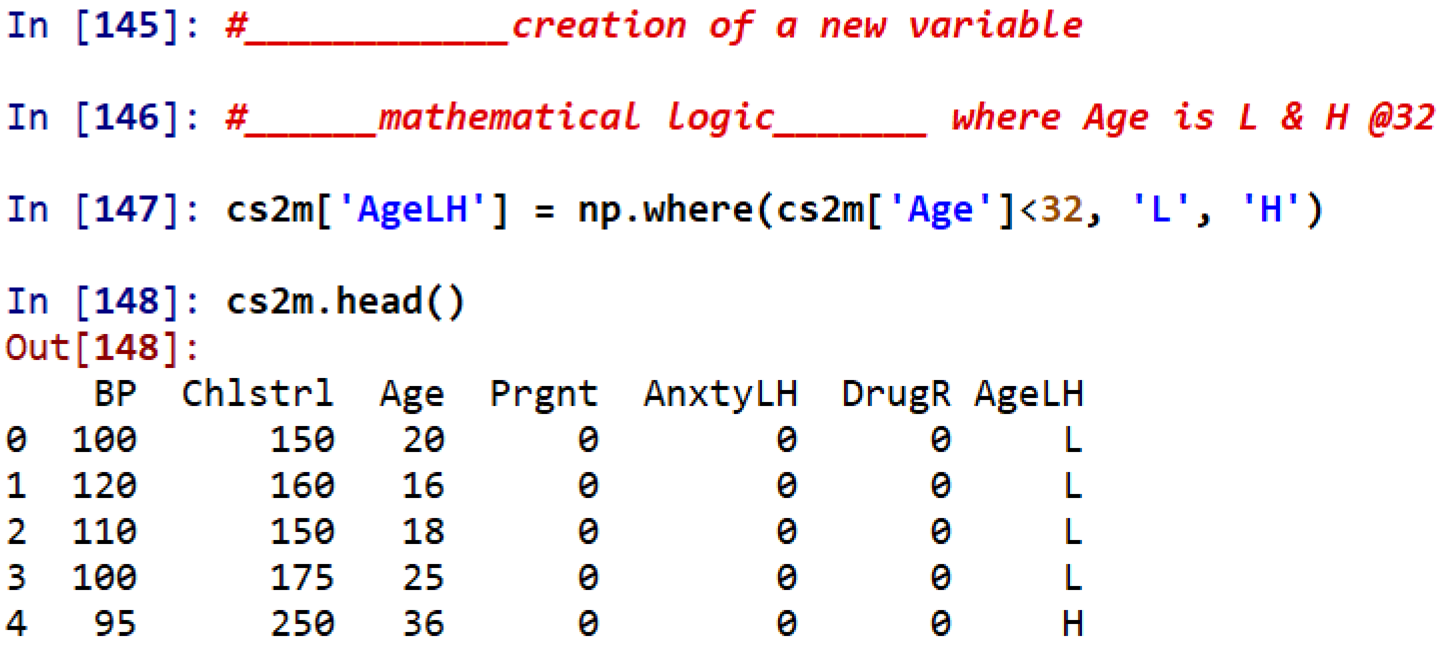
```
grades35 = pd.concat([grades3, grades5])
len(grades35.ethnicity)
grades35.head()
```



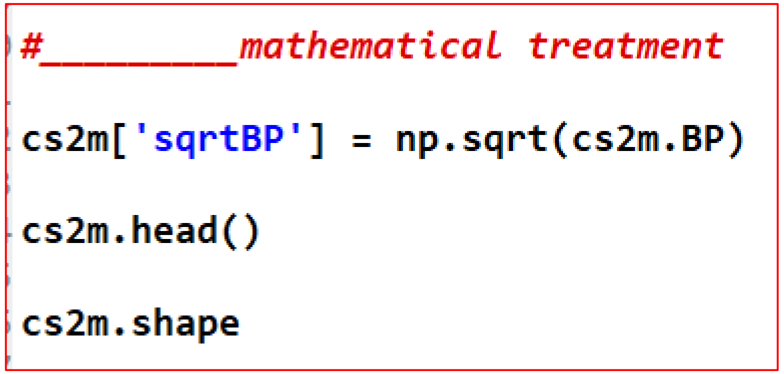
```
In [142]: grades35 = pd.concat([grades3, grades5])

In [143]: len(grades35.ethnicity)
Out[143]: 35

In [144]: grades35.head()
Out[144]:
    Sr_No      id   lastname firstname  ...  total  percent  grade  passfail
3       4  132931    OSBORNE       ANN  ...    103       82      B         P
9      10  164605   LANGFORD      DAWN  ...    124       99      A         P
24     25  302400      JONES    ROBERT  ...     65       52      F         F
25     26  307894   TORRENCE      GWEN  ...     90       72      C         P
40     41  466407  PICKERING     HEIDI  ...     84       67      D         P

[5 rows x 22 columns]
```

```
#____________creation of a new variable

#______mathematical logic_______ where Age is L & H @32

cs2m['AgeLH'] = np.where(cs2m['Age']<32, 'L', 'H')
cs2m.head()
```

New Variable

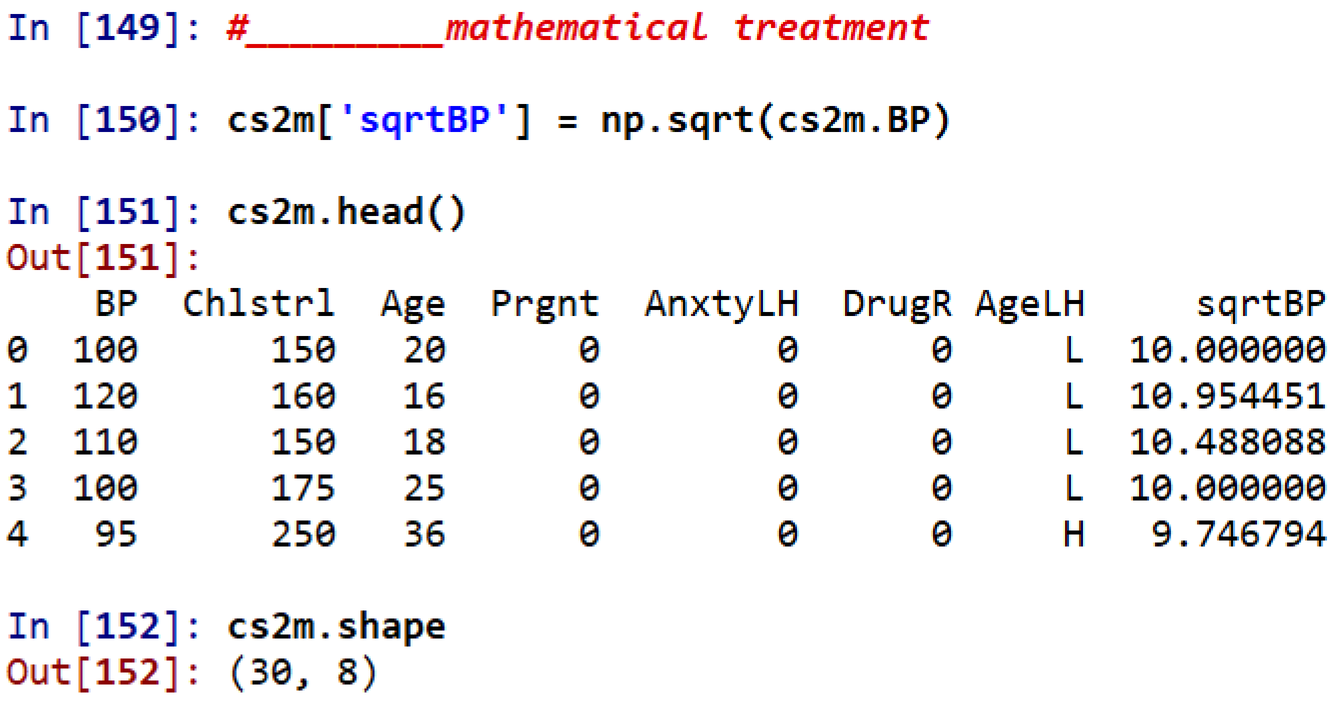


```
In [145]: #____________creation of a new variable

In [146]: #______mathematical logic_______ where Age is L & H @32

In [147]: cs2m['AgeLH'] = np.where(cs2m['Age']<32, 'L', 'H')

In [148]: cs2m.head()
Out[148]:
    BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR AgeLH
0  100      150   20      0        0      0     L
1  120      160   16      0        0      0     L
2  110      150   18      0        0      0     L
3  100      175   25      0        0      0     L
4   95      250   36      0        0      0     H
```



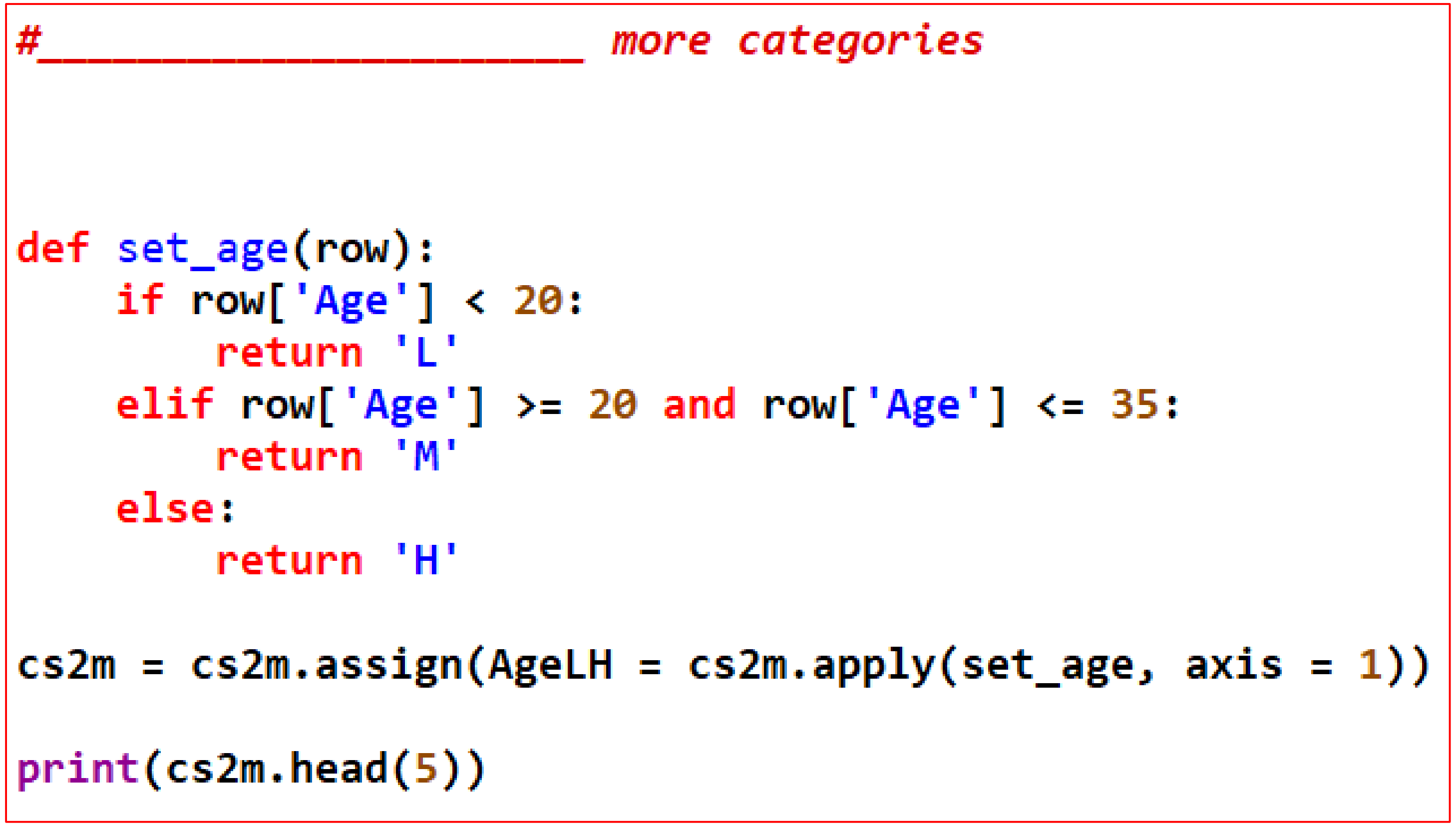
```
#_________mathematical  treatment

cs2m['sqrtBP'] = np.sqrt(cs2m.BP)

cs2m.head()

cs2m.shape
```

New Variable



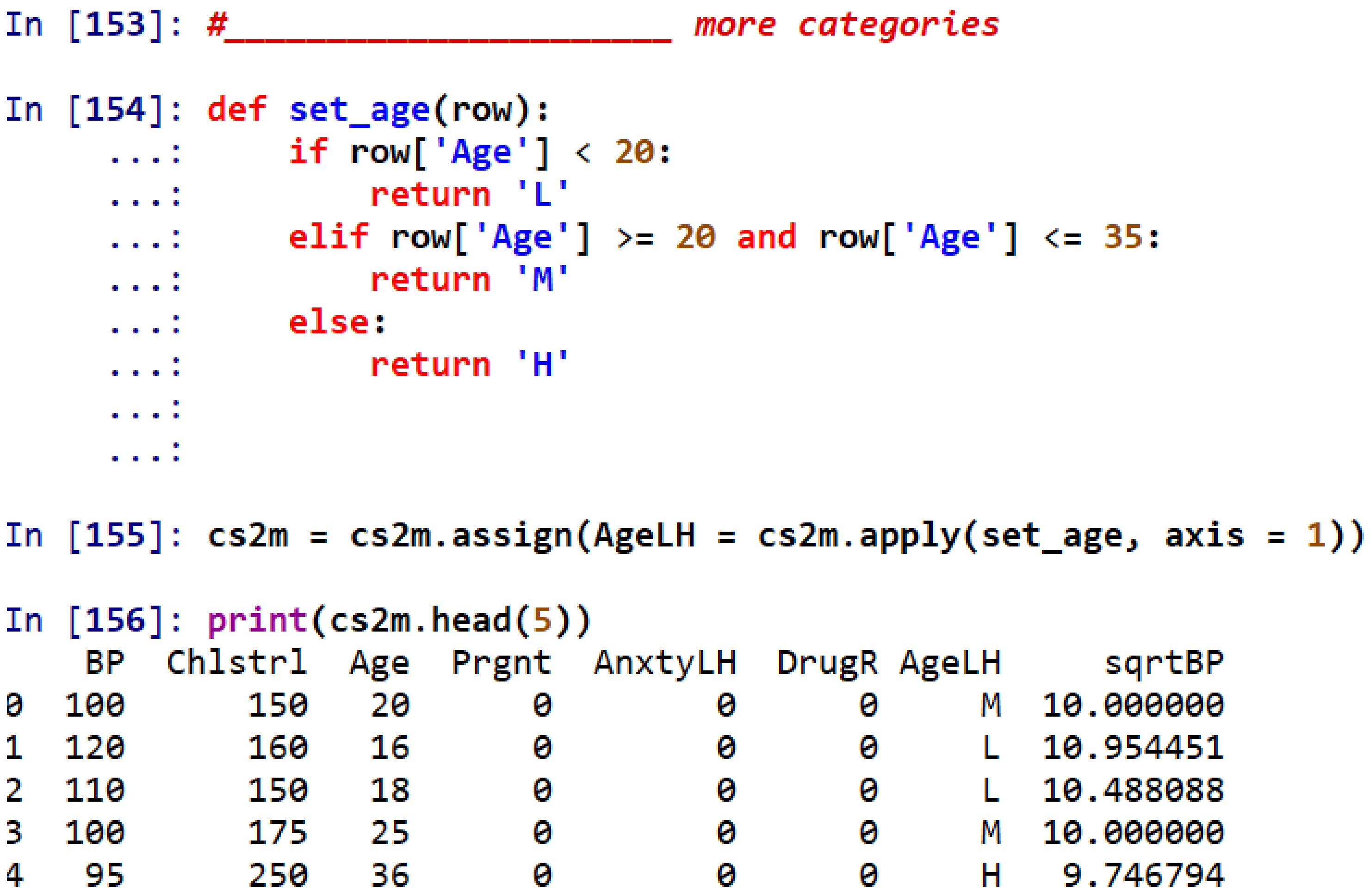
```
In [149]: #_________mathematical  treatment

In [150]: cs2m['sqrtBP'] = np.sqrt(cs2m.BP)

In [151]: cs2m.head()
Out[151]:
    BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR AgeLH     sqrtBP
0  100      150   20      0        0      0     L  10.000000
1  120      160   16      0        0      0     L  10.954451
2  110      150   18      0        0      0     L  10.488088
3  100      175   25      0        0      0     L  10.000000
4   95      250   36      0        0      0     H   9.746794

In [152]: cs2m.shape
Out[152]: (30, 8)
```

New Variable



```
#_______________________ more categories



def set_age(row):
    if row['Age'] < 20:
        return 'L'
    elif row['Age'] >= 20 and row['Age'] <= 35:
        return 'M'
    else:
        return 'H'

cs2m = cs2m.assign(AgeLH = cs2m.apply(set_age, axis = 1))

print(cs2m.head(5))
```

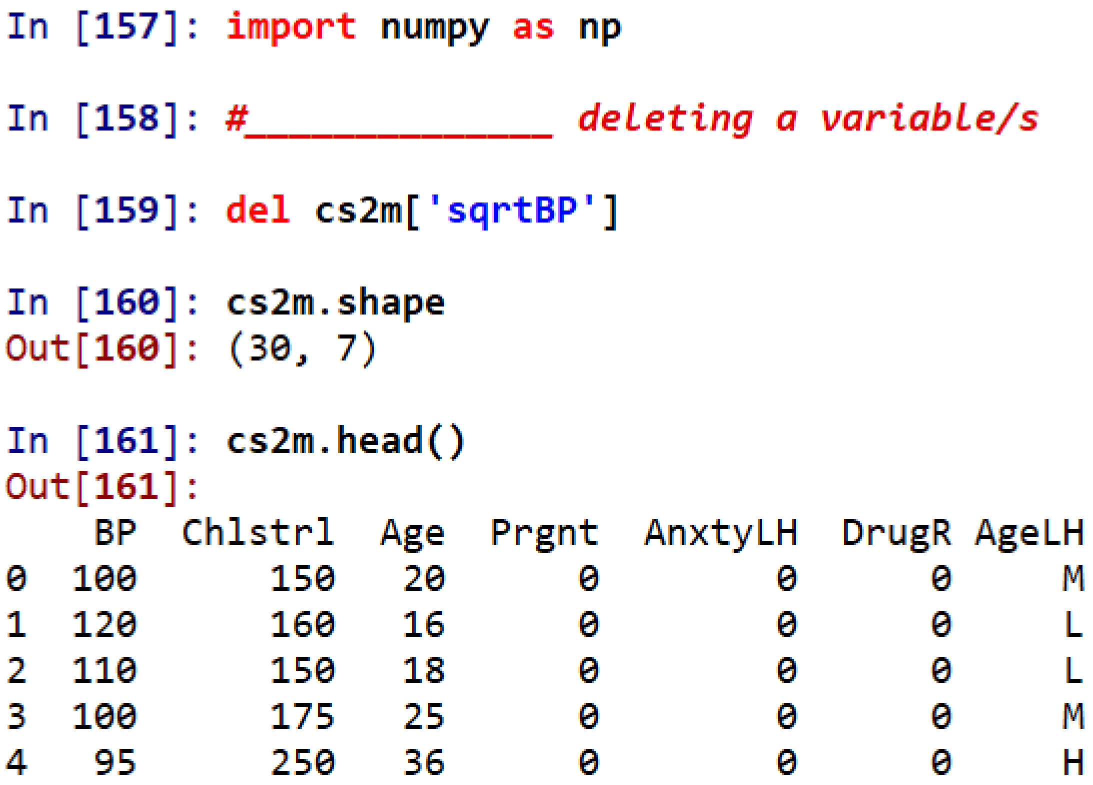
New Variable



```
In [153]: #______________________ more categories

In [154]: def set_age(row):
     ...:     if row['Age'] < 20:
     ...:         return 'L'
     ...:     elif row['Age'] >= 20 and row['Age'] <= 35:
     ...:         return 'M'
     ...:     else:
     ...:         return 'H'
     ...:
     ...:

In [155]: cs2m = cs2m.assign(AgeLH = cs2m.apply(set_age, axis = 1))

In [156]: print(cs2m.head(5))
    BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR AgeLH     sqrtBP
0  100      150   20      0        0      0     M  10.000000
1  120      160   16      0        0      0     L  10.954451
2  110      150   18      0        0      0     L  10.488088
3  100      175   25      0        0      0     M  10.000000
4   95      250   36      0        0      0     H   9.746794
```

Remove Variable

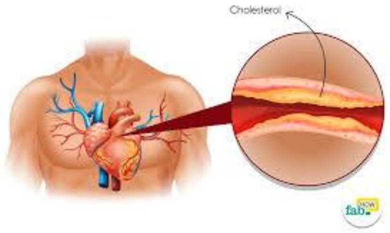
```
import numpy as np

#______________ deleting a variable/s

del cs2m['sqrtBP']
cs2m.shape
cs2m.head()
```



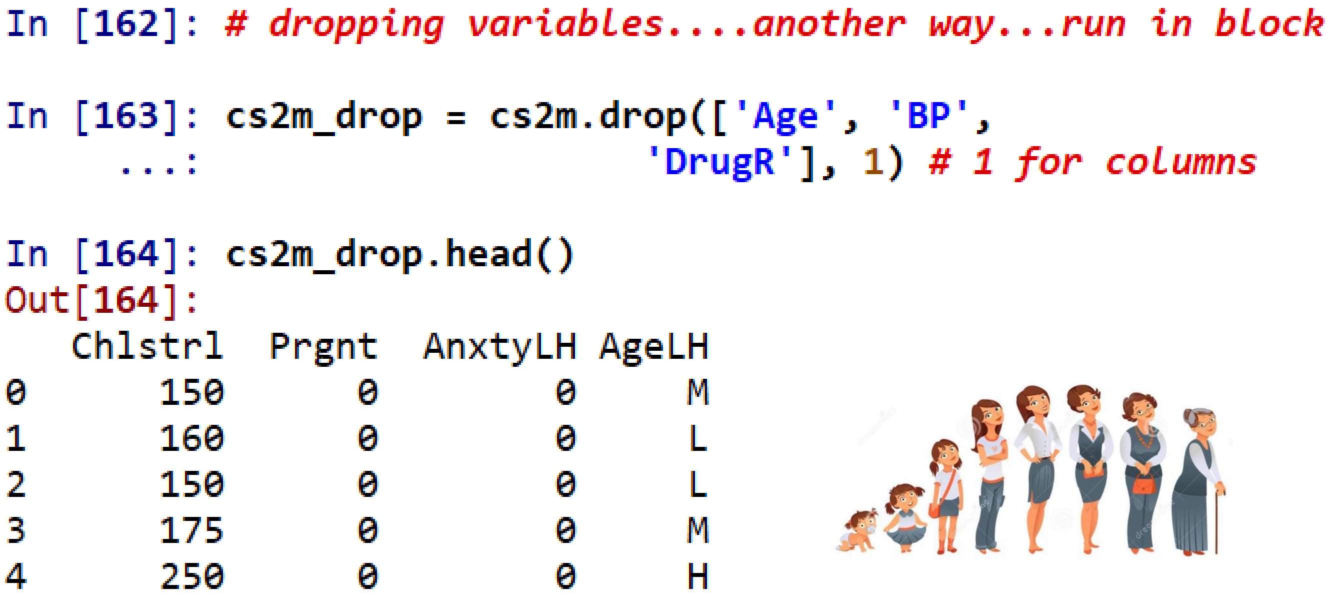
```
In [157]: import numpy as np

In [158]: #______________ deleting a variable/s

In [159]: del cs2m['sqrtBP']

In [160]: cs2m.shape
Out[160]: (30, 7)

In [161]: cs2m.head()
Out[161]:
    BP  Chlstrl  Age  Prgnt  AnxtyLH  DrugR AgeLH
0  100      150   20      0        0      0     M
1  120      160   16      0        0      0     L
2  110      150   18      0        0      0     L
3  100      175   25      0        0      0     M
4   95      250   36      0        0      0     H
```

```
# dropping variables....another way...run in block

cs2m_drop = cs2m.drop(['Age', 'BP',
                       'DrugR'], 1) # 1 for columns

cs2m_drop.head()
```

Remove Variable

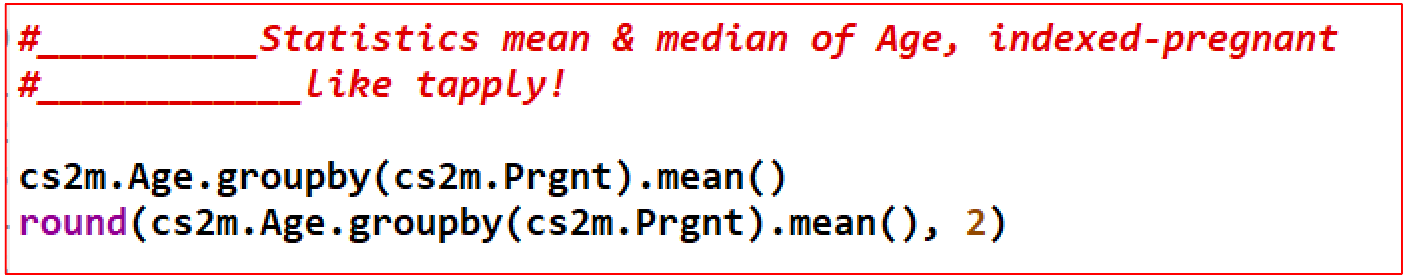


rstock.com • 1074559082



```
In [162]: # dropping variables....another way...run in block

In [163]: cs2m_drop = cs2m.drop(['Age', 'BP',
     ...:                        'DrugR'], 1) # 1 for columns

In [164]: cs2m_drop.head()
Out[164]:
   Chlstrl  Prgnt  AnxtyLH AgeLH
0      150      0        0     M
1      160      0        0     L
2      150      0        0     L
3      175      0        0     M
4      250      0        0     H
```

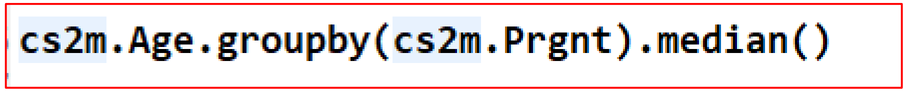
Statistics across a categorical variable



```
#__________Statistics mean & median of Age, indexed-pregnant
#____________like tapply!

cs2m.Age.groupby(cs2m.Prgnt).mean()
round(cs2m.Age.groupby(cs2m.Prgnt).mean(), 2)
```

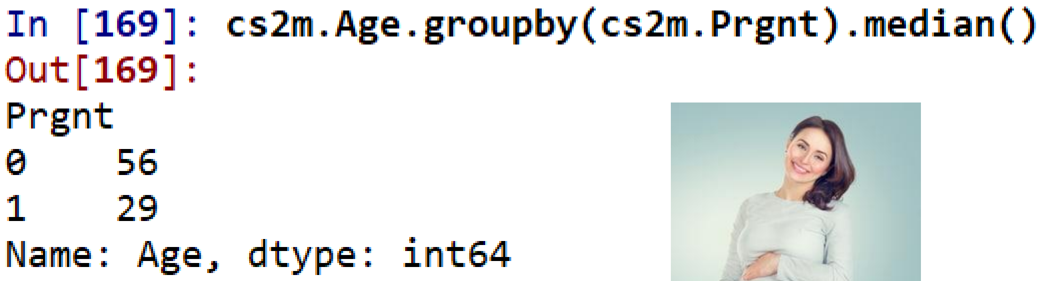
```
In [165]: #__________Statistics mean & median of Age, indexed-pregnant

In [166]: #____________like tapply!

In [167]: cs2m.Age.groupby(cs2m.Prgnt).mean()
Out[167]:
Prgnt
0    48.000000
1    27.533333
Name: Age, dtype: float64

In [168]: round(cs2m.Age.groupby(cs2m.Prgnt).mean(), 2)
Out[168]:
Prgnt
0    48.00
1    27.53
Name: Age, dtype: float64
```

ɪrstock.com • 1074559082



```
cs2m.Age.groupby(cs2m.Prgnt).median()
```

Statistics across a categorical variable

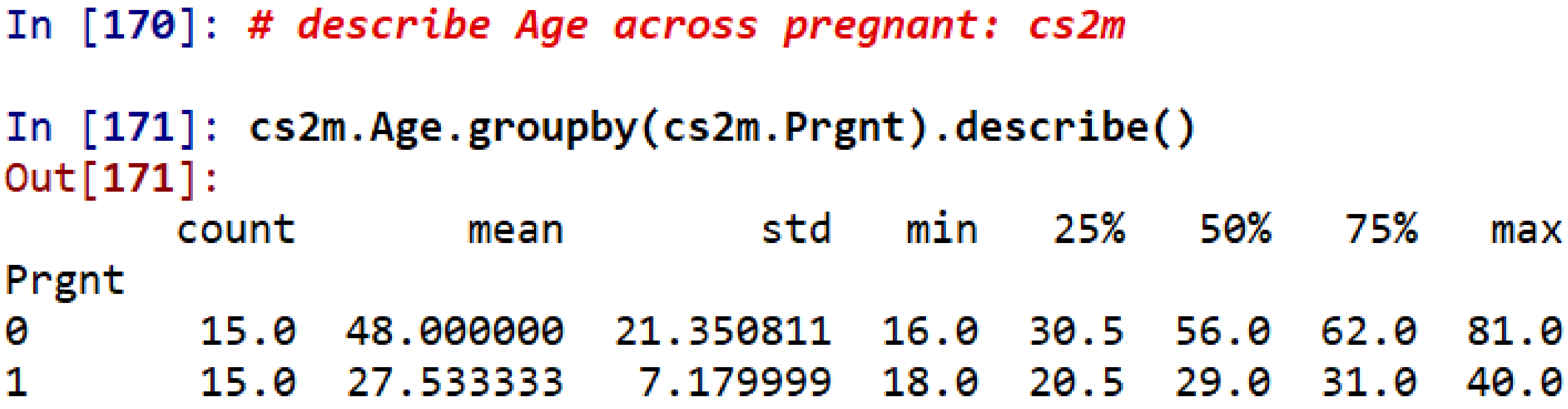


```
In [169]: cs2m.Age.groupby(cs2m.Prgnt).median()
Out[169]:
Prgnt
0    56
1    29
Name: Age, dtype: int64
```

ːrstock.com • 1074559082

Statistics across a categorical variable

```
# describe Age across pregnant: cs2m

cs2m.Age.groupby(cs2m.Prgnt).describe()
```



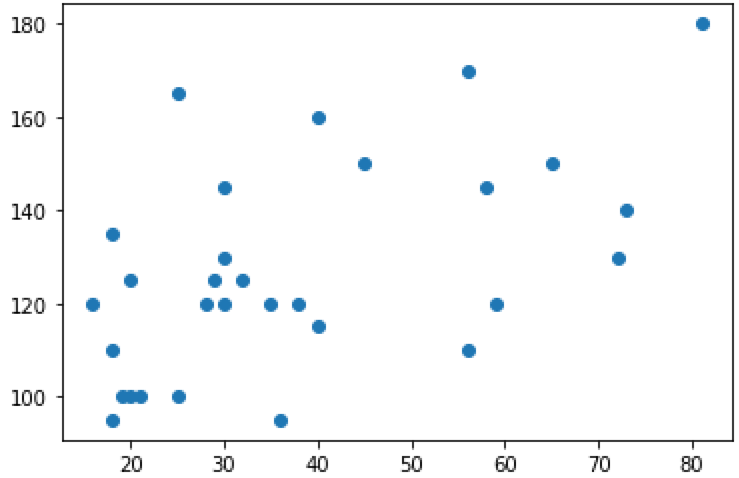
```
In [170]: # describe Age across pregnant: cs2m

In [171]: cs2m.Age.groupby(cs2m.Prgnt).describe()
Out[171]:
       count       mean        std   min   25%   50%   75%   max
Prgnt
0       15.0  48.000000  21.350811  16.0  30.5  56.0  62.0  81.0
1       15.0  27.533333   7.179999  18.0  20.5  29.0  31.0  40.0
```

ɪrstock.com • 1074559082

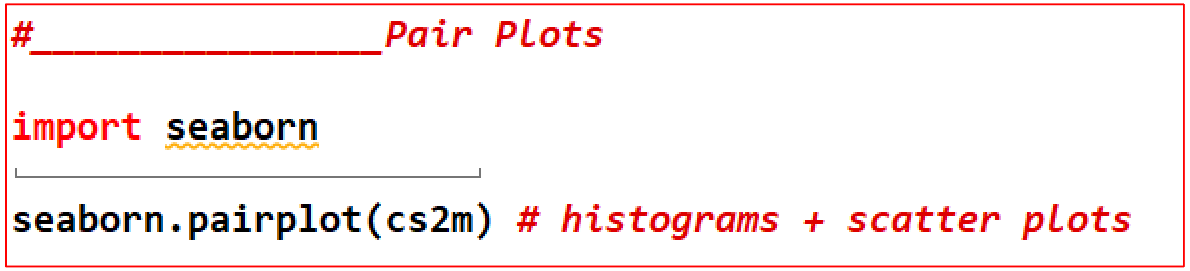
```
#_______________scatter plots

plt.scatter(cs2m['Age'], cs2m[['BP']])
# as excel, 1st will form X-Axis
```

Scatter Plot

```
In [172]: #_______________scatter plots

In [173]: plt.scatter(cs2m['Age'], cs2m[['BP']])
Out[173]: <matplotlib.collections.PathCollection at 0x22aa6248198>
```

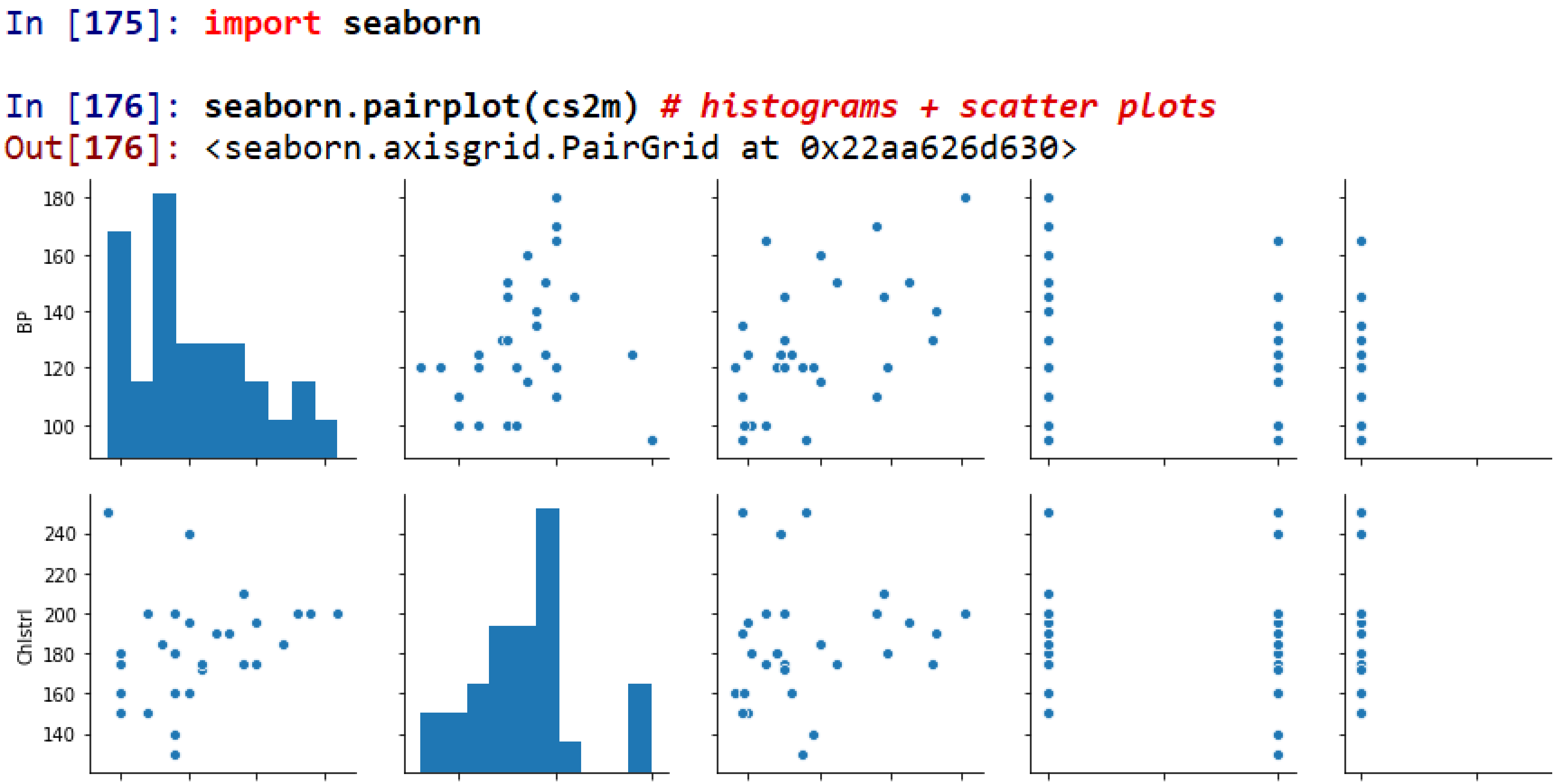


```
In [174]: # as excel, 1st will form X-Axis
```

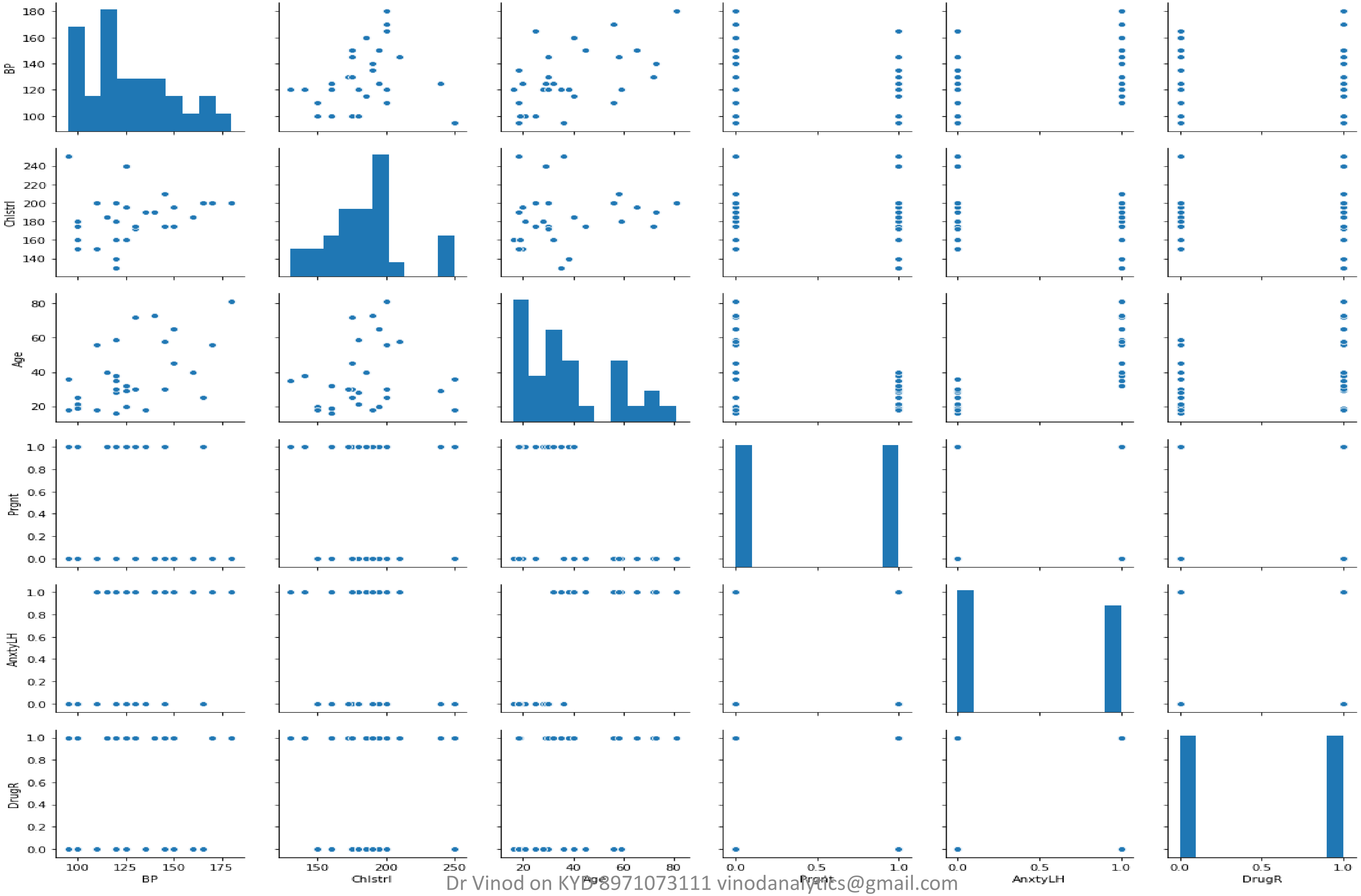


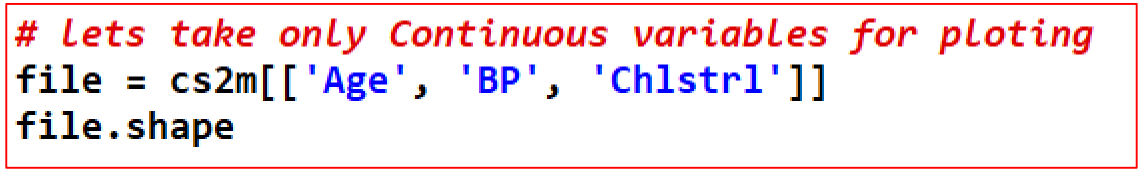
```
#________________Pair Plots

import seaborn

seaborn.pairplot(cs2m) # histograms + scatter plots
```

Pair Plots

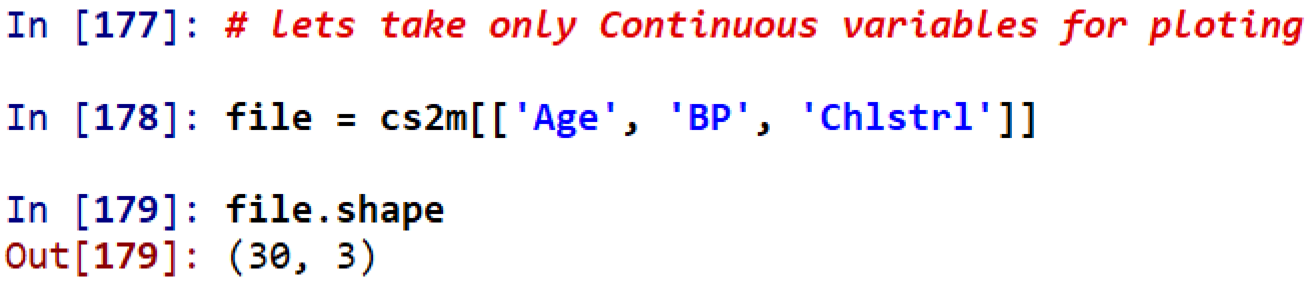


```
In [175]: import seaborn

In [176]: seaborn.pairplot(cs2m) # histograms + scatter plots
Out[176]: <seaborn.axisgrid.PairGrid at 0x22aa626d630>
```

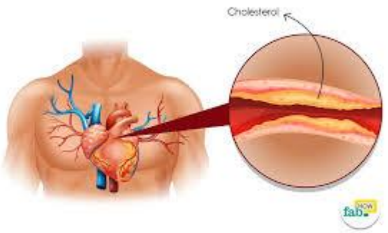

BP
Chlstrl
Age
Prgnt
AnxtyLH
DrugR
180
160
140
120
100
240
220
200
180
160
140
80
60
40
20
1.0
0.8
0.6
0.4
0.2
0.0
100
125
150
175
150
200
250
20
40
60
80
0.0
0.5
1.0



```
# lets take only Continuous variables for ploting
file = cs2m[['Age', 'BP', 'Chlstrl']]
file.shape
```

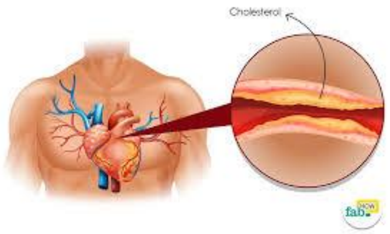


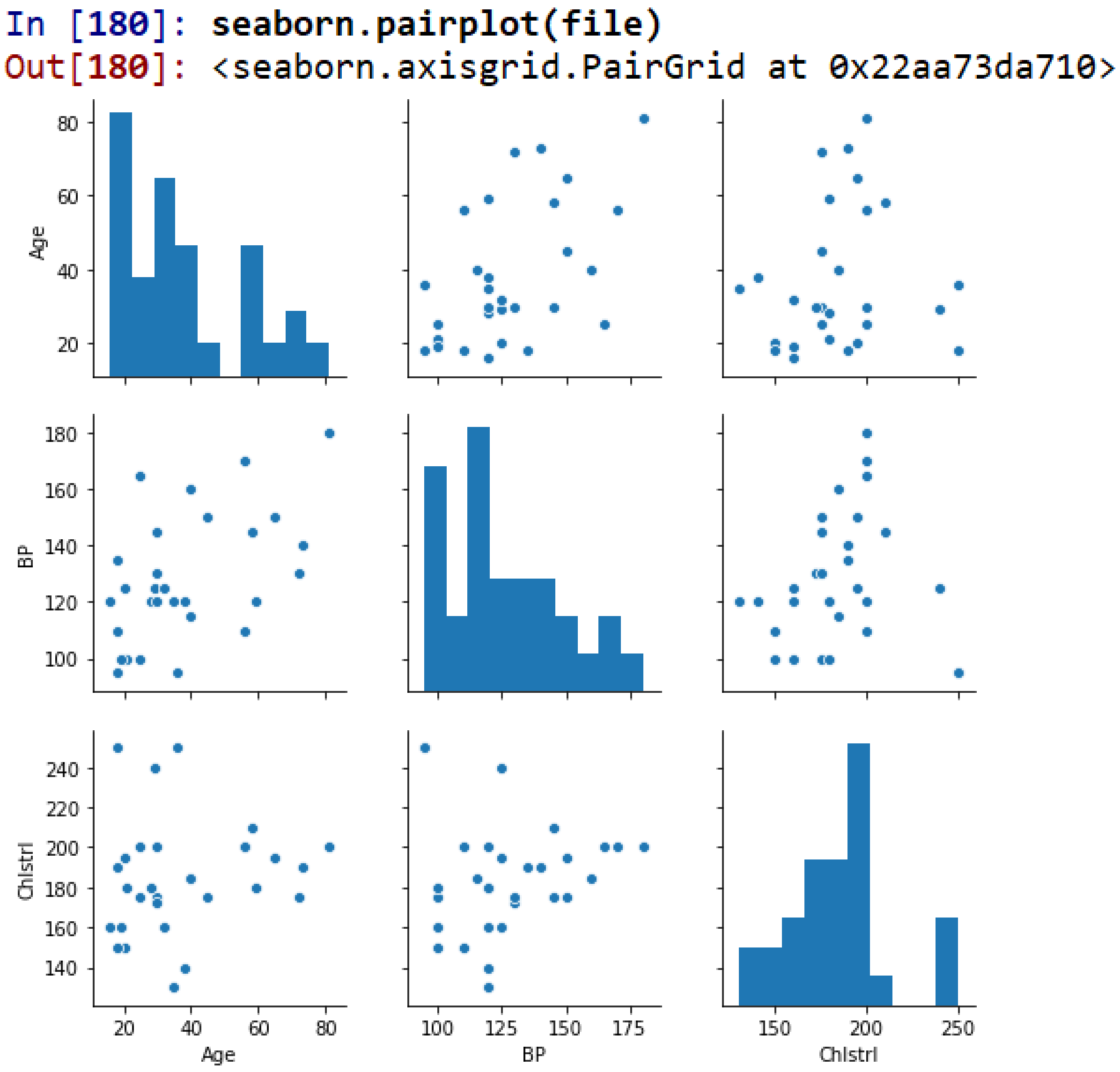
```
In [177]: # lets take only Continuous variables for ploting

In [178]: file = cs2m[['Age', 'BP', 'Chlstrl']]

In [179]: file.shape
Out[179]: (30, 3)
```



```
seaborn.pairplot(file)
```

Pair Plots



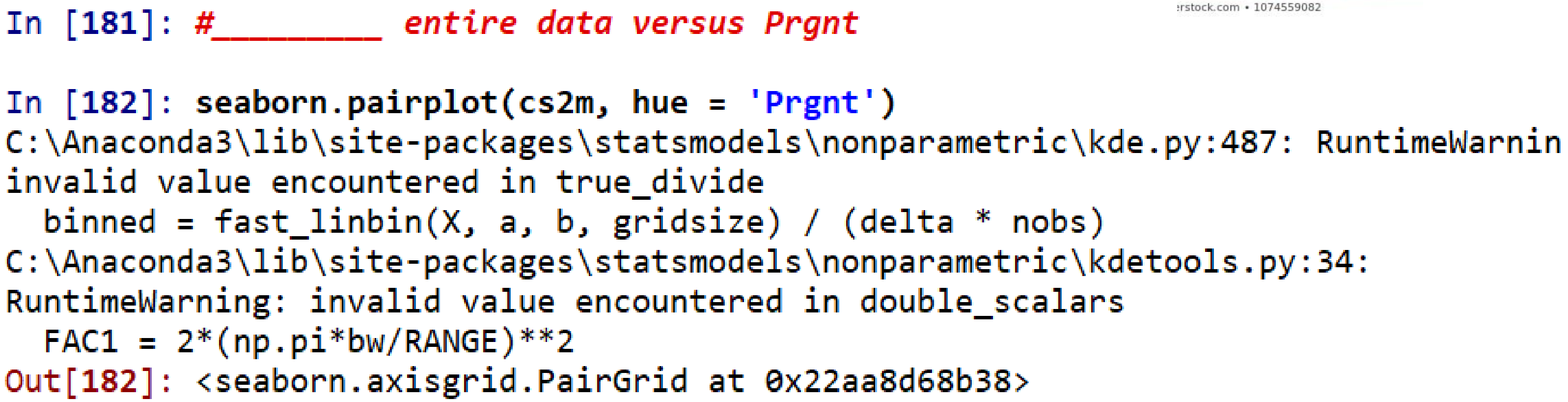


```
In [180]: seaborn.pairplot(file)
Out[180]: <seaborn.axisgrid.PairGrid at 0x22aa73da710>
```

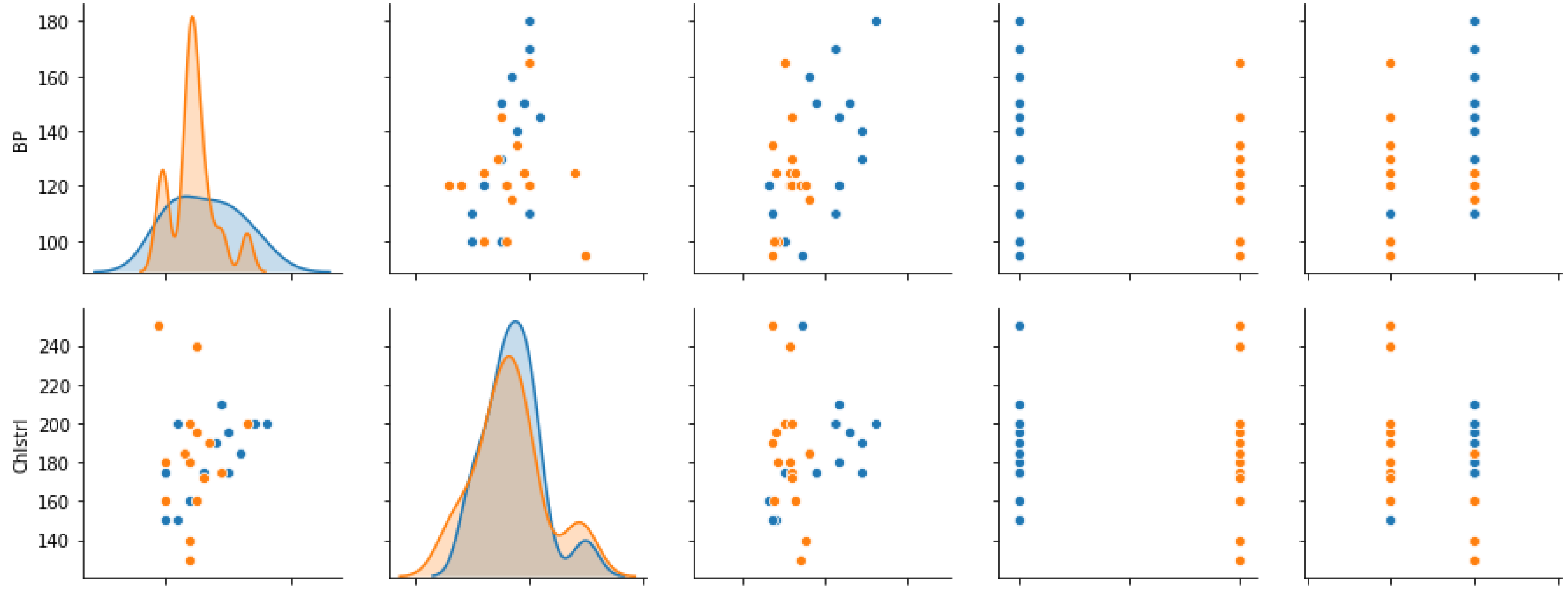
```
#_________ entire data versus Prgnt

seaborn.pairplot(cs2m, hue = 'Prgnt')

# density plots + scatter plots
```

Pair Plots

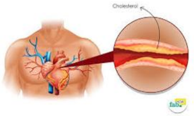
rstock.com • 1074559082

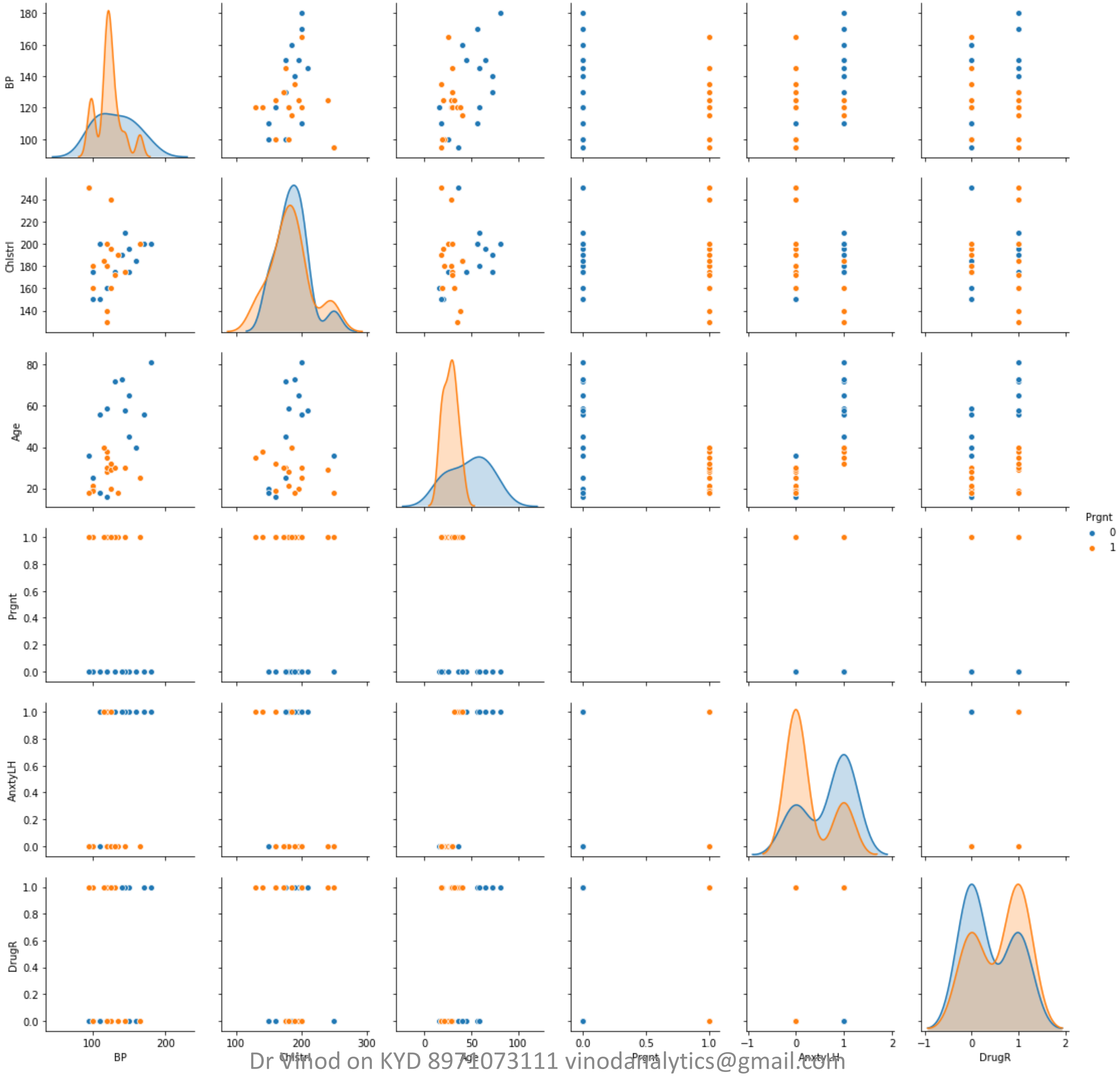


```
In [181]: #_________ entire data versus Prgnt

In [182]: seaborn.pairplot(cs2m, hue = 'Prgnt')
C:\Anaconda3\lib\site-packages\statsmodels\nonparametric\kde.py:487: RuntimeWarnin
invalid value encountered in true_divide
  binned = fast_linbin(X, a, b, gridsize) / (delta * nobs)
C:\Anaconda3\lib\site-packages\statsmodels\nonparametric\kdetools.py:34:
RuntimeWarning: invalid value encountered in double_scalars
  FAC1 = 2*(np.pi*bw/RANGE)**2
Out[182]: <seaborn.axisgrid.PairGrid at 0x22aa8d68b38>
```



Pair Plots

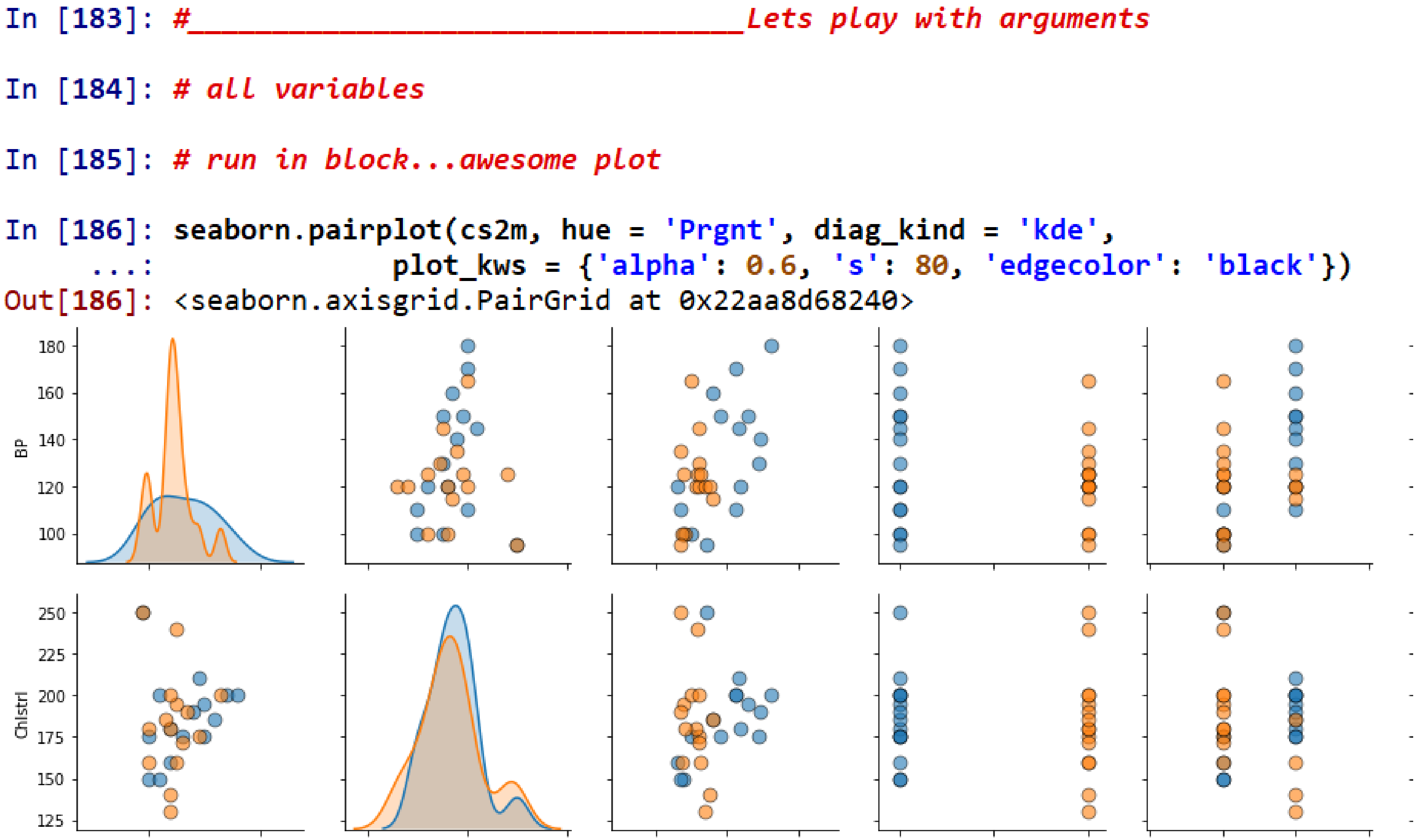




ɪrstock.com • 1074559082

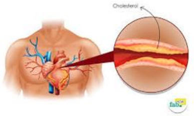
```
#_________________________________Lets play with arguments

# all variables
# run in block...awesome plot
seaborn.pairplot(cs2m, hue = 'Prgnt', diag_kind = 'kde',
             plot_kws = {'alpha': 0.6, 's': 80, 'edgecolor': 'black'})
```

Pair Plots

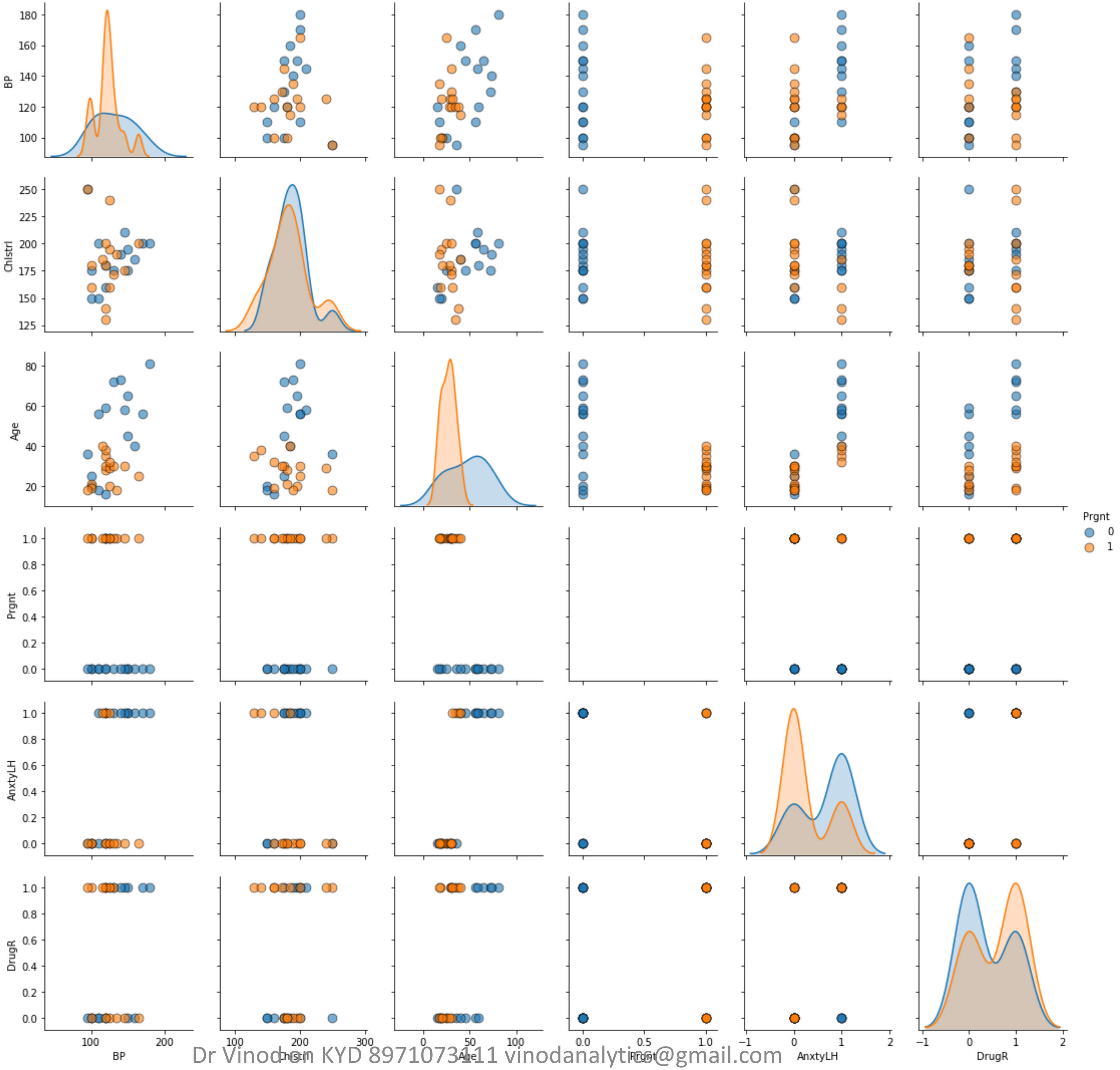


```
In [183]: #_________________________________Lets play with arguments

In [184]: # all variables

In [185]: # run in block...awesome plot

In [186]: seaborn.pairplot(cs2m, hue = 'Prgnt', diag_kind = 'kde',
     ...:             plot_kws = {'alpha': 0.6, 's': 80, 'edgecolor': 'black'})
Out[186]: <seaborn.axisgrid.PairGrid at 0x22aa8d68240>
```

Pair Plots



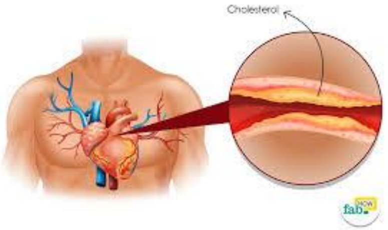
Worry
Anxiety
Fear
Panic
Tension


BP
Chlstrl
Age
Prgnt
AnxtyLH
DrugR
180
160
140
120
100
250
225
200
175
150
125
80
60
40
20
1.0
0.8
0.6
0.4
0.2
0.0
100
200
300
0
50
0.5
−1
1
2
Prgnt
0
1

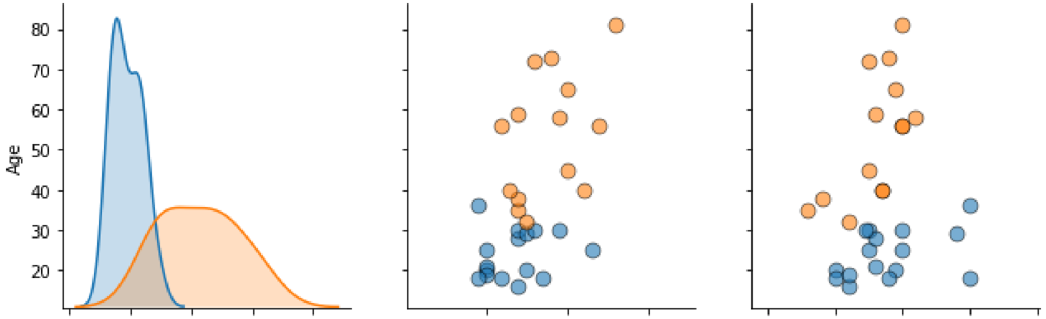
rstock.com • 1074559082

```
#___________ SELECTED variable
# run in block...awesome plot

seaborn.pairplot(cs2m,
             vars = ['Age', 'BP', 'Chlstrl'],
             hue = 'AnxtyLH', diag_kind = 'kde',
             plot_kws = {'alpha': 0.6, 's': 80, 'edgecolor': 'black'},
             size = 3)
```

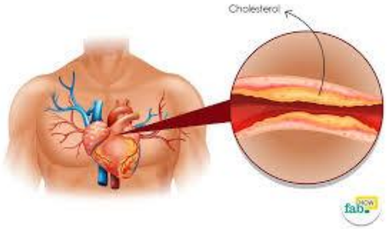
Pair Plots

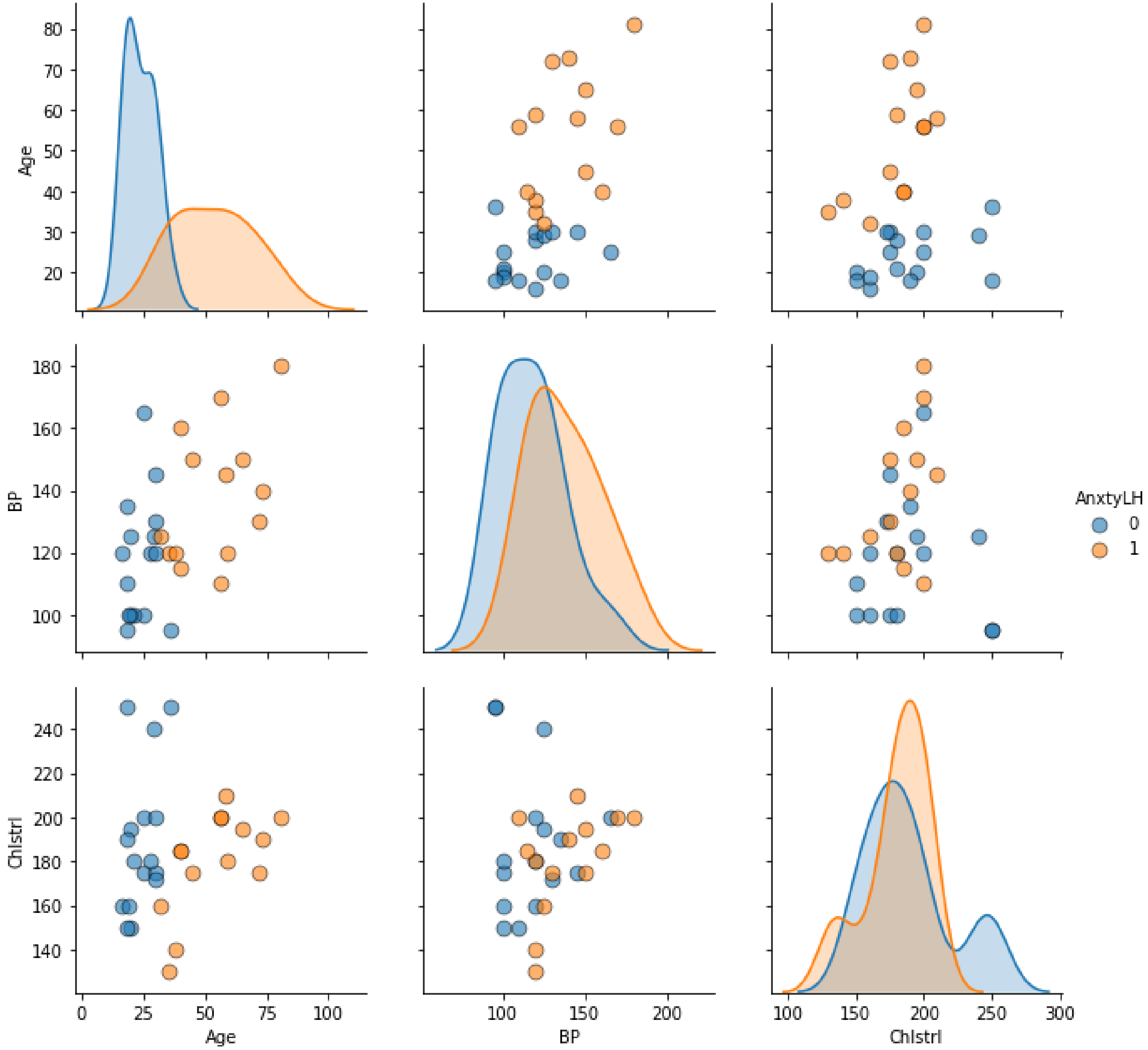


```
In [187]: #___________ SELECTED variable

In [188]: # run in block...awesome plot

In [189]: seaborn.pairplot(cs2m,
     ...:                 vars = ['Age', 'BP', 'Chlstrl'],
     ...:                 hue = 'AnxtyLH', diag_kind = 'kde',
     ...:                 plot_kws = {'alpha': 0.6, 's': 80, 'edgecolor': 'black'},
     ...:                 size = 3)
C:\Anaconda3\lib\site-packages\seaborn\axisgrid.py:2065: UserWarning: The `size`
parameter has been renamed to `height`; pleaes update your code.
  warnings.warn(msg, UserWarning)
Out[189]: <seaborn.axisgrid.PairGrid at 0x22aacaf9e10>
```



Pair Plots

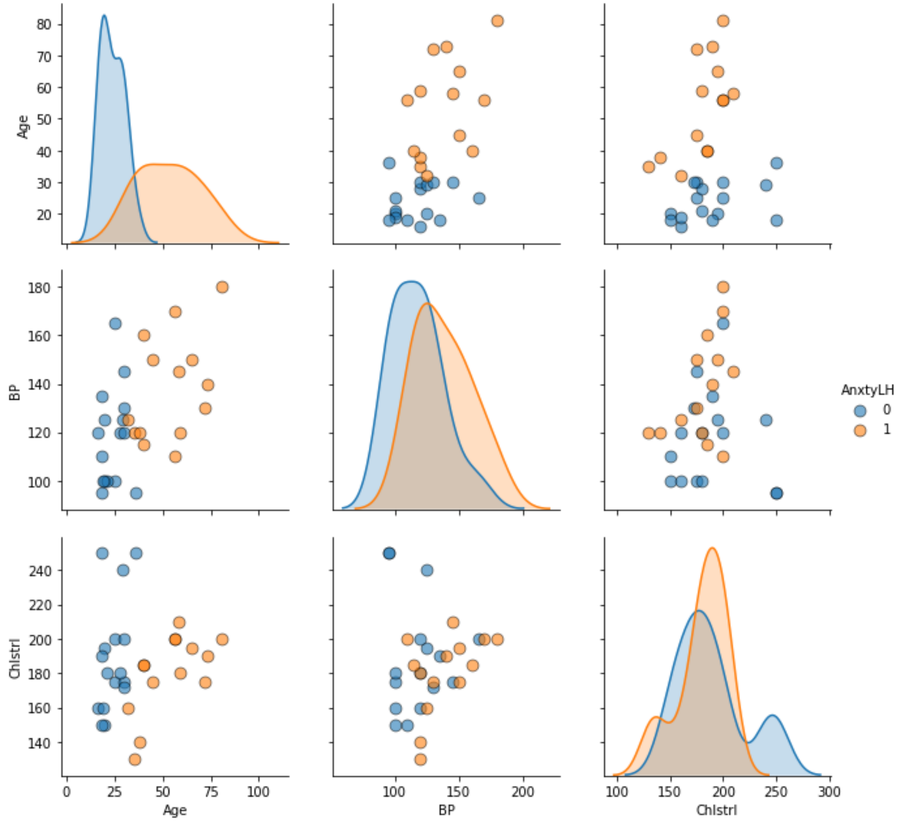

Cholesterol

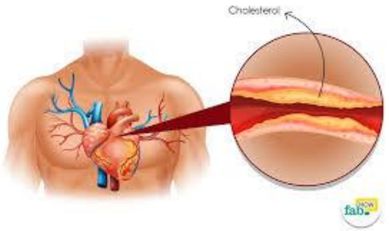

Age
80
70
60
50
40
30
20
BP
180
160
140
120
100
Chlstrl
240
220
200
180
160
140
0
25
50
75
100
Age
100
150
200
BP
100
150
200
250
300
Chlstrl
AnxtyLH
0
1

Worry
Anxiety
Fear
Panic
Tension

```
# change values of alpha (transperancy, 0 to 1, 1 for highest visibility)
# keep s as 10 to 80, size of circles
# edgecolor (boundary of circles) is black in most cases, try green
# size is overall size of plot, try from 1 to 6
```

```
#___________ SELECTED variable
# run in block...awesome plot

seaborn.pairplot(cs2m,
                 vars = ['Age', 'BP', 'Chlstrl'],
                 hue = 'AnxtyLH', diag_kind = 'kde',
                 plot_kws = {'alpha': 0.6, 's': 80, 'edgecolor': 'black'},
                 size = 3)
```

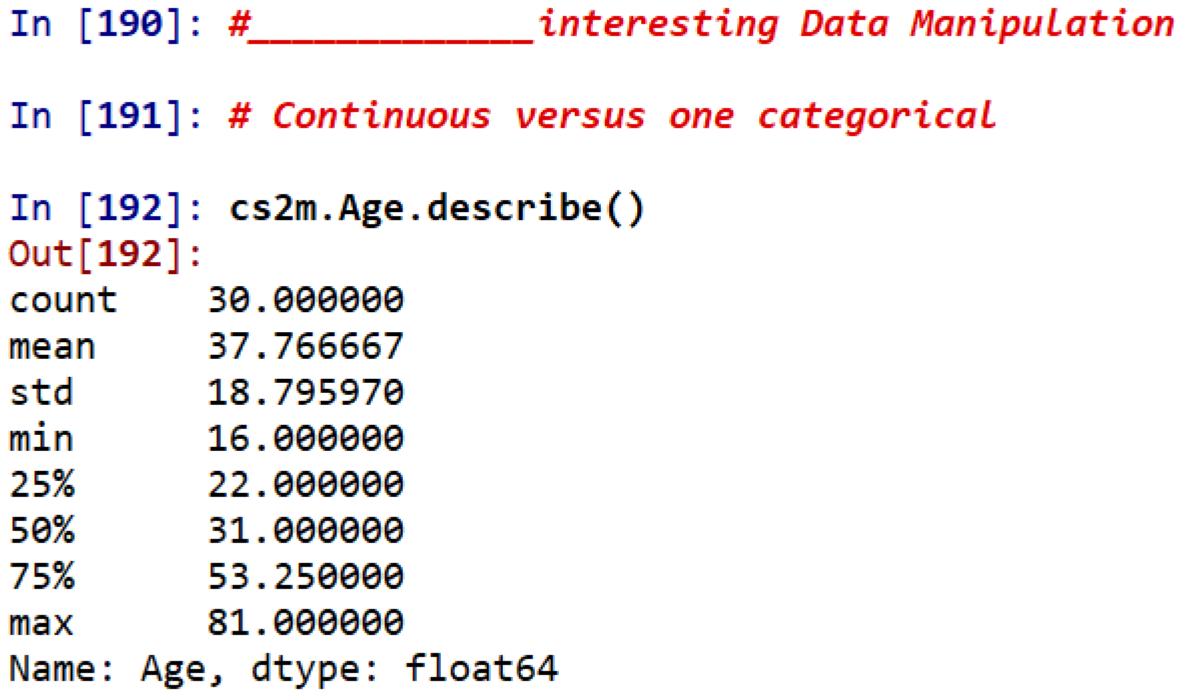




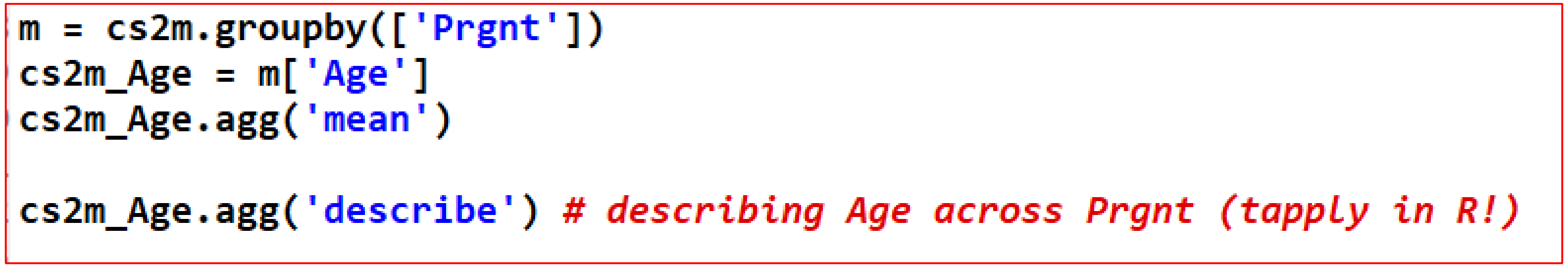
Pair Plots

```
#_____________interesting Data Manipulation

# Continuous versus one categorical
cs2m.Age.describe()
```

Statistics:
One Continuous Variable

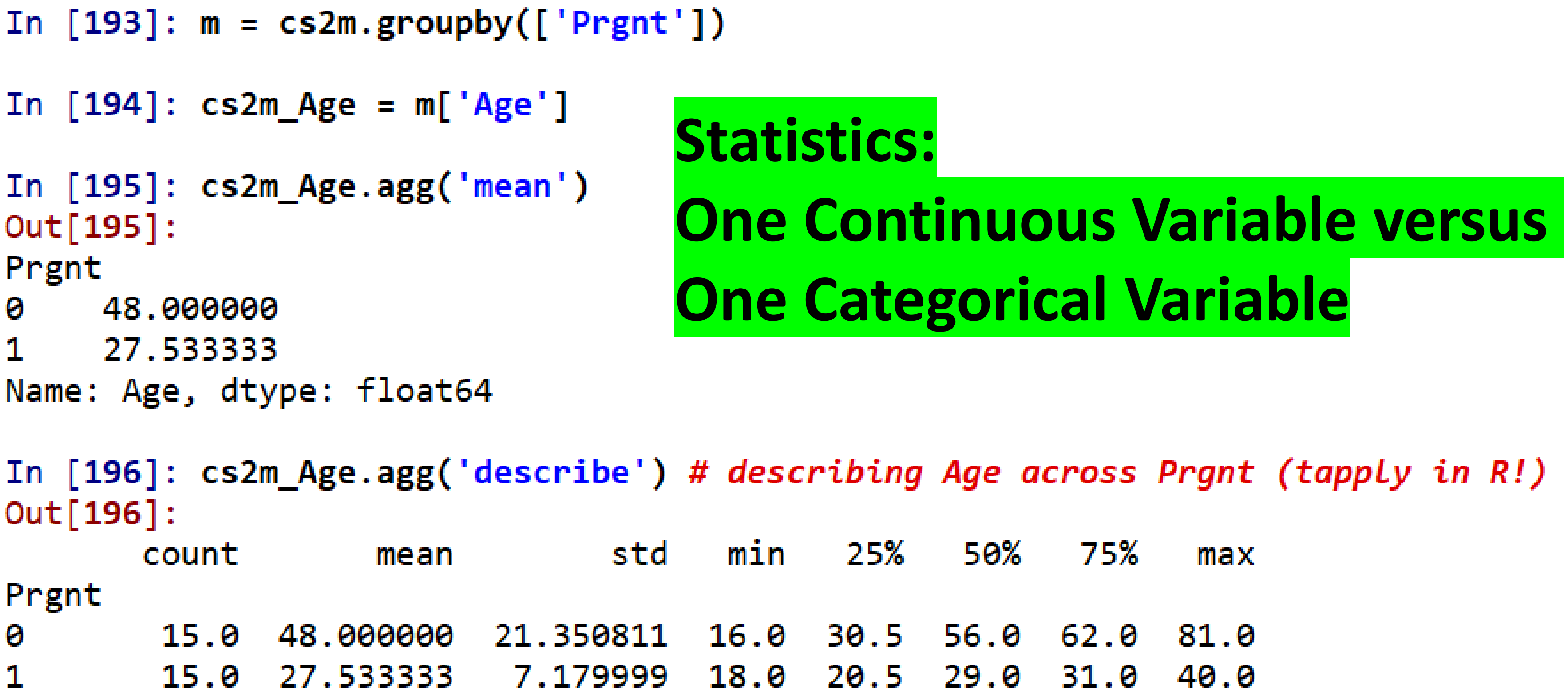


```
In [190]: #_____________interesting Data Manipulation

In [191]: # Continuous versus one categorical

In [192]: cs2m.Age.describe()
Out[192]:
count    30.000000
mean     37.766667
std      18.795970
min      16.000000
25%      22.000000
50%      31.000000
75%      53.250000
max      81.000000
Name: Age, dtype: float64
```



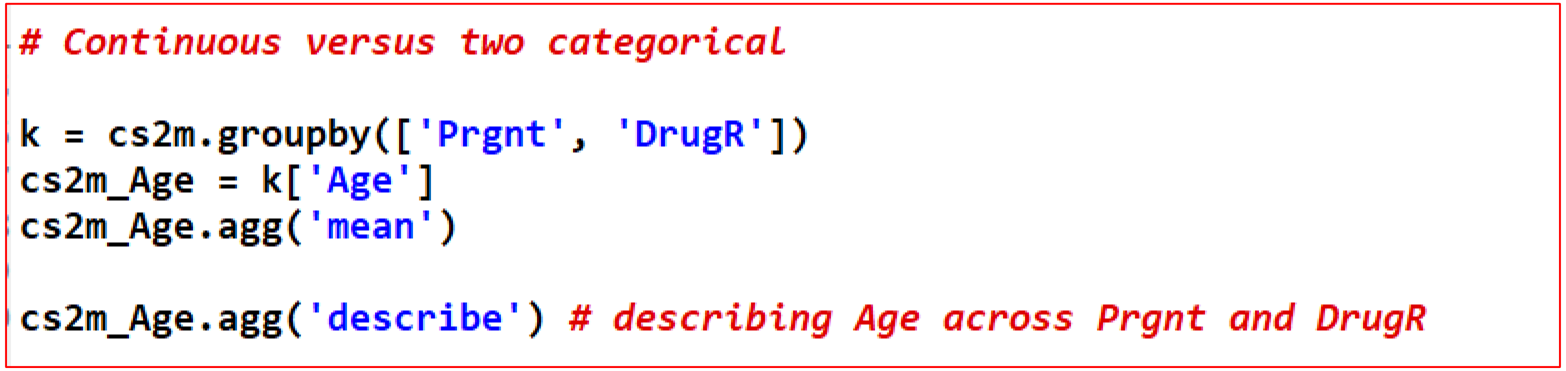
```
m = cs2m.groupby(['Prgnt'])
cs2m_Age = m['Age']
cs2m_Age.agg('mean')

cs2m_Age.agg('describe') # describing Age across Prgnt (tapply in R!)
```

Statistics:
One Continuous Variable versus One Categorical Variable



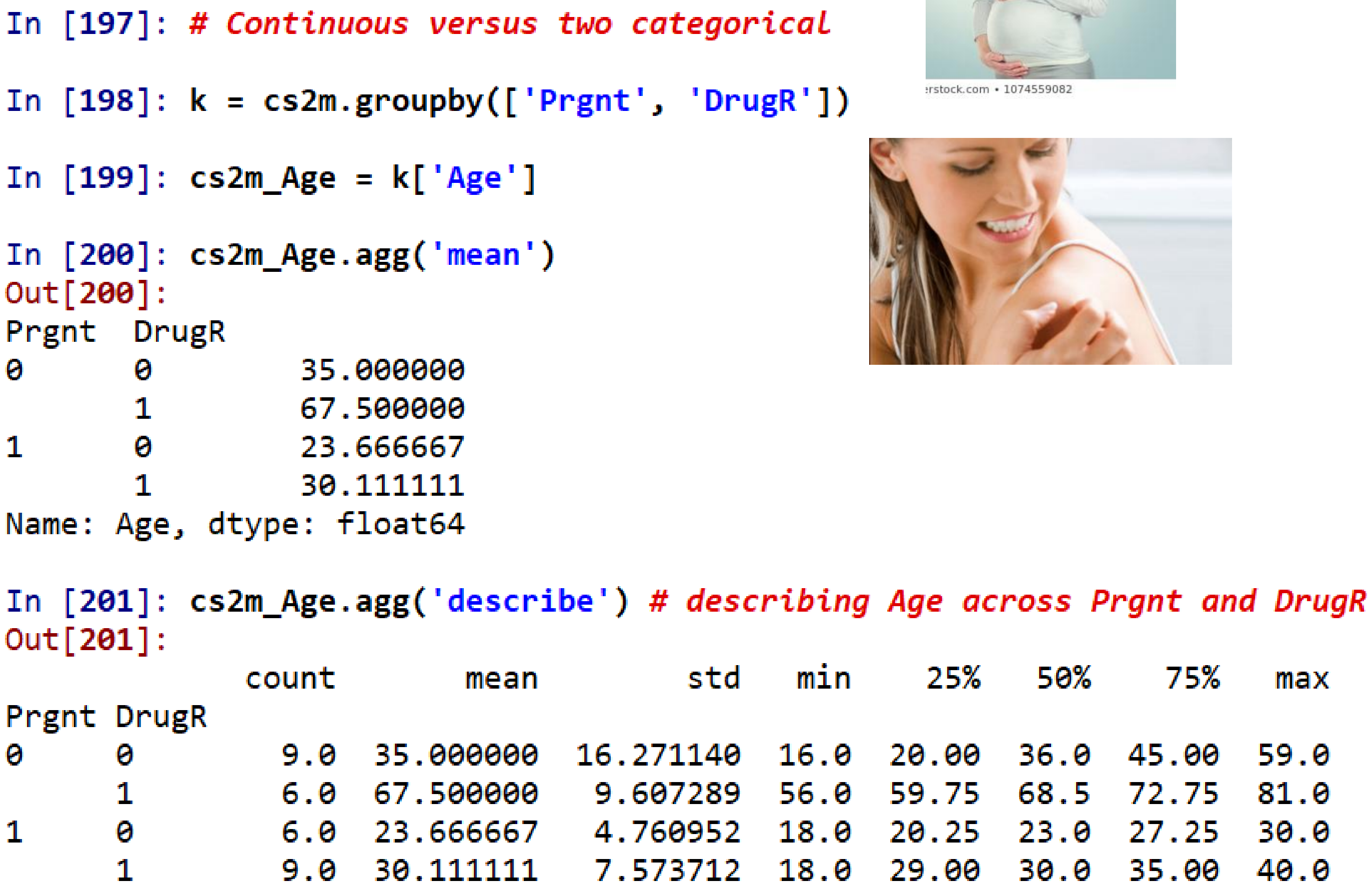
```
In [193]: m = cs2m.groupby(['Prgnt'])

In [194]: cs2m_Age = m['Age']

In [195]: cs2m_Age.agg('mean')
Out[195]:
Prgnt
0    48.000000
1    27.533333
Name: Age, dtype: float64

In [196]: cs2m_Age.agg('describe') # describing Age across Prgnt (tapply in R!)
Out[196]:
       count       mean        std   min   25%   50%   75%   max
Prgnt
0       15.0  48.000000  21.350811  16.0  30.5  56.0  62.0  81.0
1       15.0  27.533333   7.179999  18.0  20.5  29.0  31.0  40.0
```

rstock.com • 1074559082



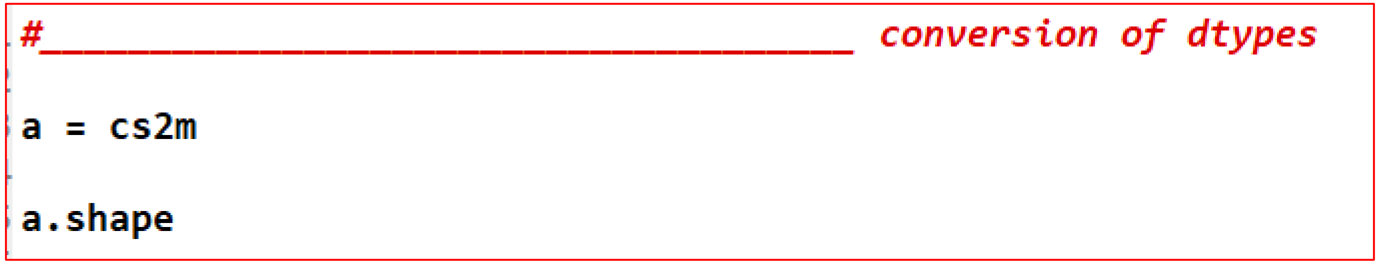
```
# Continuous versus two categorical

k = cs2m.groupby(['Prgnt', 'DrugR'])
cs2m_Age = k['Age']
cs2m_Age.agg('mean')

cs2m_Age.agg('describe') # describing Age across Prgnt and DrugR
```

Statistics:
One Continuous Variable versus Two Categorical Variables

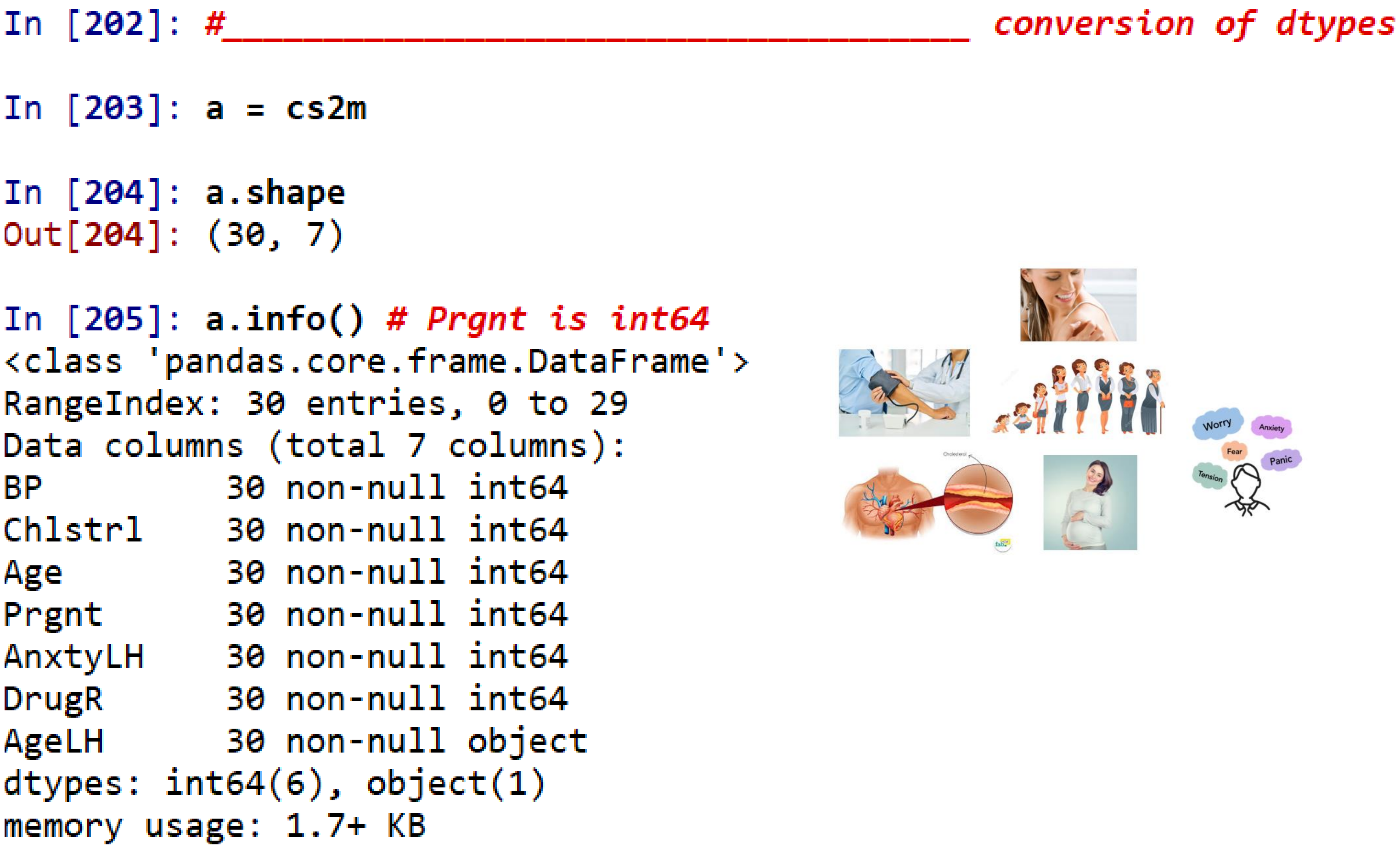


```
In [197]: # Continuous versus two categorical

In [198]: k = cs2m.groupby(['Prgnt', 'DrugR'])

In [199]: cs2m_Age = k['Age']

In [200]: cs2m_Age.agg('mean')
Out[200]:
Prgnt  DrugR
0      0        35.000000
       1        67.500000
1      0        23.666667
       1        30.111111
Name: Age, dtype: float64

In [201]: cs2m_Age.agg('describe') # describing Age across Prgnt and DrugR
Out[201]:
             count       mean        std   min    25%   50%    75%   max
Prgnt DrugR
0     0        9.0  35.000000  16.271140  16.0  20.00  36.0  45.00  59.0
      1        6.0  67.500000   9.607289  56.0  59.75  68.5  72.75  81.0
1     0        6.0  23.666667   4.760952  18.0  20.25  23.0  27.25  30.0
      1        9.0  30.111111   7.573712  18.0  29.00  30.0  35.00  40.0
```

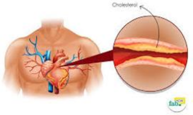
rstock.com • 1074559082

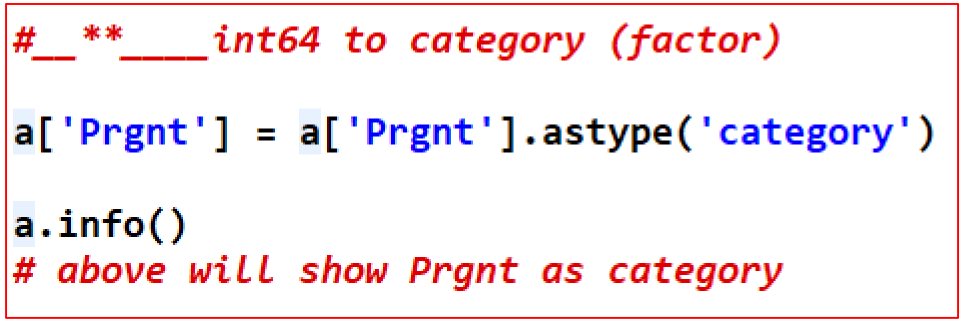


```
#______________________________________ conversion of dtypes

a = cs2m

a.shape
```



```
In [202]: #______________________________________ conversion of dtypes

In [203]: a = cs2m

In [204]: a.shape
Out[204]: (30, 7)

In [205]: a.info() # Prgnt is int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 7 columns):
BP         30 non-null int64
Chlstrl    30 non-null int64
Age        30 non-null int64
Prgnt      30 non-null int64
AnxtyLH    30 non-null int64
DrugR      30 non-null int64
AgeLH      30 non-null object
dtypes: int64(6), object(1)
memory usage: 1.7+ KB
```





```
#__**____int64 to category (factor)

a['Prgnt'] = a['Prgnt'].astype('category')

a.info()
# above will show Prgnt as category
```

rstock.com • 1074559082

```
In [206]: #__**____int64 to category (factor)

In [207]: a['Prgnt'] = a['Prgnt'].astype('category')

In [208]: a.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 7 columns):
BP          30 non-null int64
Chlstrl     30 non-null int64
Age         30 non-null int64
Prgnt       30 non-null category
AnxtyLH     30 non-null int64
DrugR       30 non-null int64
AgeLH       30 non-null object
dtypes: category(1), int64(5), object(1)
memory usage: 1.6+ KB

In [209]: # above will show Prgnt as category
```

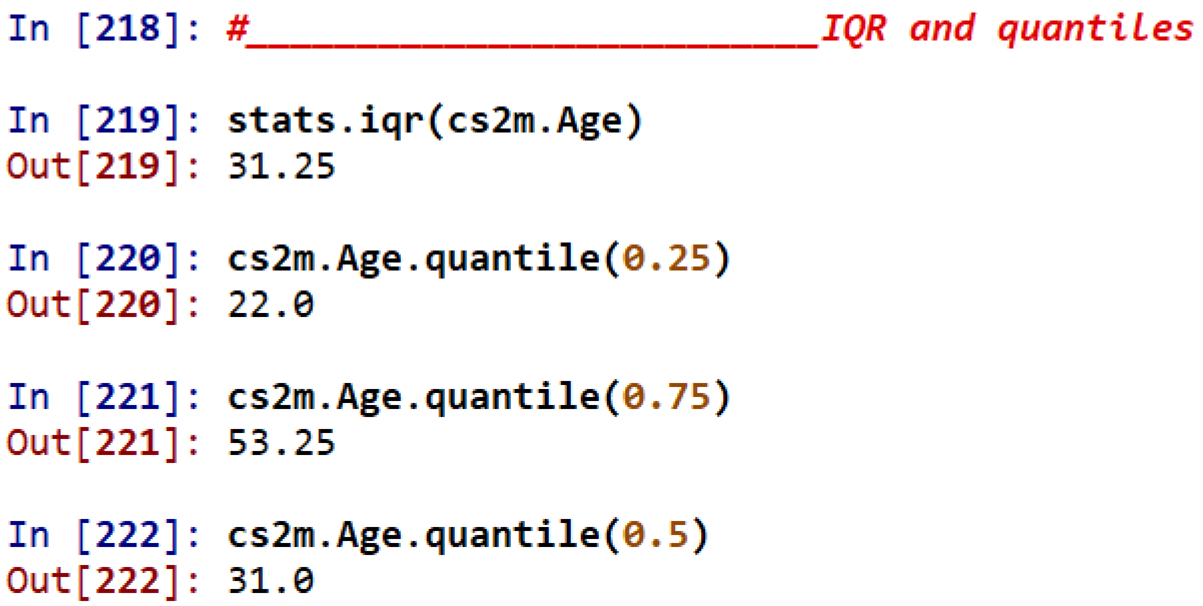
```
#__**____int64 to float (numeric)

a['Age'] = a['Age'].astype('float')
a.info()
# above has changed Age to float
```

```
In [210]: #__**____int64 to float (numeric)

In [211]: a['Age'] = a['Age'].astype('float')

In [212]: a.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 7 columns):
BP          30 non-null int64
Chlstrl     30 non-null int64
Age         30 non-null float64
Prgnt       30 non-null category
AnxtyLH     30 non-null int64
DrugR       30 non-null int64
AgeLH       30 non-null object
dtypes: category(1), float64(1), int64(4), object(1)
memory usage: 1.6+ KB

In [213]: # above has changed Age to float
```

```
#__**____back to int64 (integer)

a['Age'] = a['Age'].astype('int64')
a.info()
# above changed Age to int
```

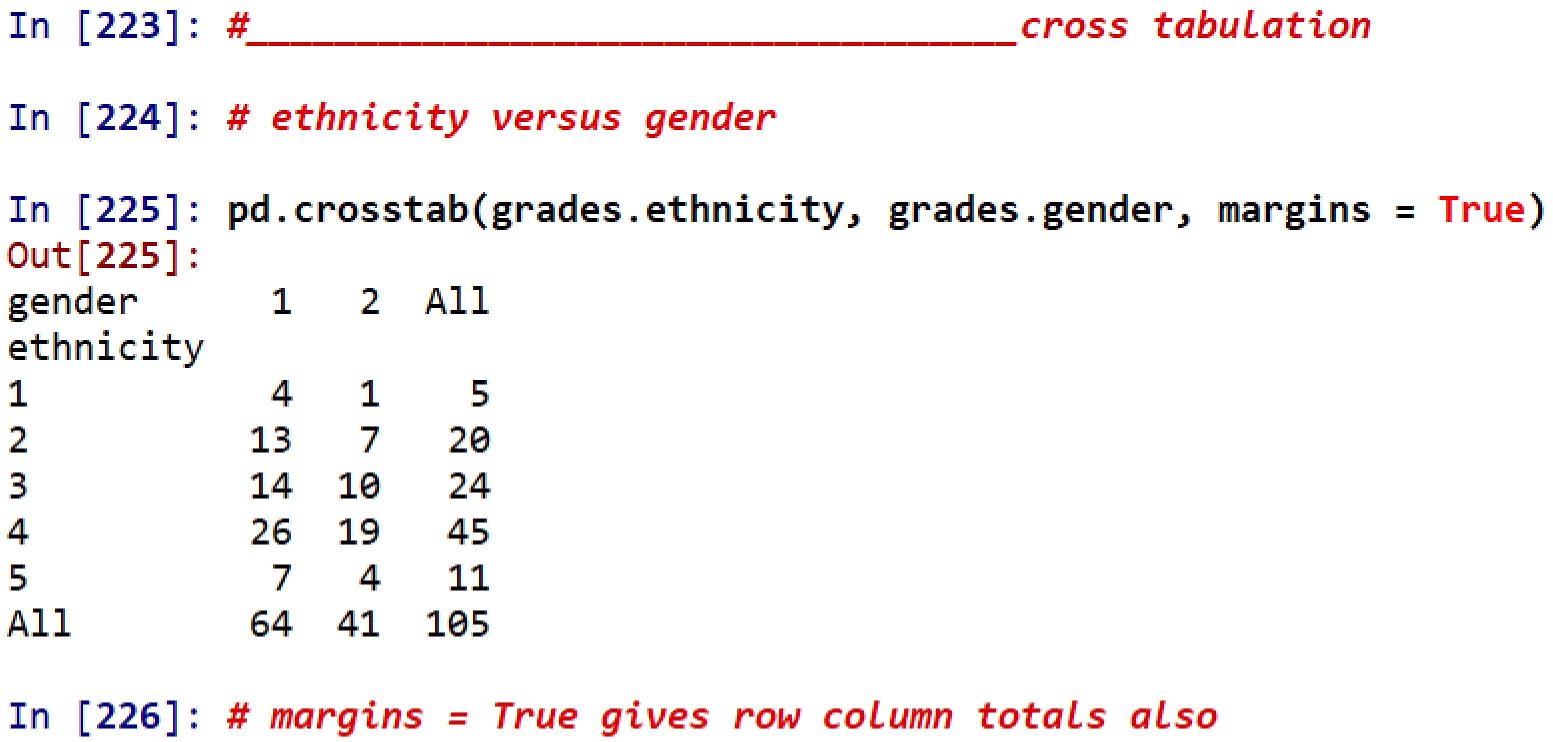
```
In [214]: #__**____back to int64 (integer)

In [215]: a['Age'] = a['Age'].astype('int64')

In [216]: a.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 7 columns):
BP          30 non-null int64
Chlstrl     30 non-null int64
Age         30 non-null int64
Prgnt       30 non-null category
AnxtyLH     30 non-null int64
DrugR       30 non-null int64
AgeLH       30 non-null object
dtypes: category(1), int64(5), object(1)
memory usage: 1.6+ KB

In [217]: # above changed Age to int
```

```
#____________________________IQR and quantiles
stats.iqr(cs2m.Age)

cs2m.Age.quantile(0.25)

cs2m.Age.quantile(0.75)

cs2m.Age.quantile(0.5)
```



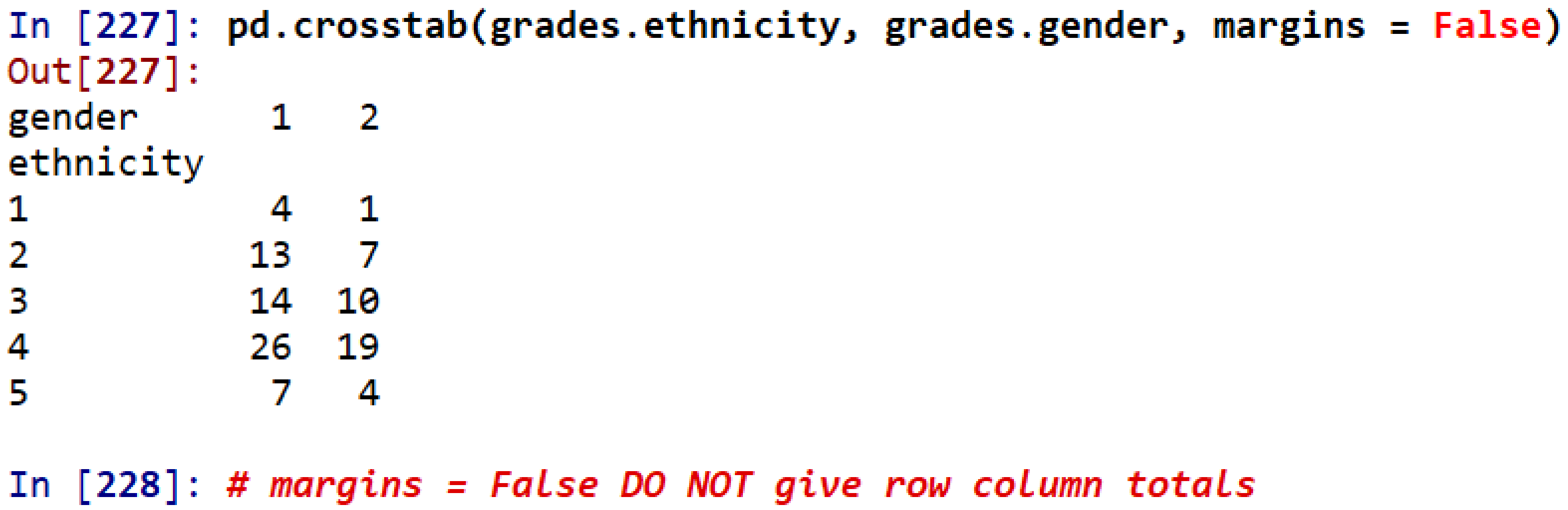
```
In [218]: #____________________________IQR and quantiles

In [219]: stats.iqr(cs2m.Age)
Out[219]: 31.25

In [220]: cs2m.Age.quantile(0.25)
Out[220]: 22.0

In [221]: cs2m.Age.quantile(0.75)
Out[221]: 53.25

In [222]: cs2m.Age.quantile(0.5)
Out[222]: 31.0
```

```
#___________________________________cross tabulation

# ethnicity versus gender

pd.crosstab(grades.ethnicity, grades.gender, margins = True)
# margins = True gives row column totals also
```

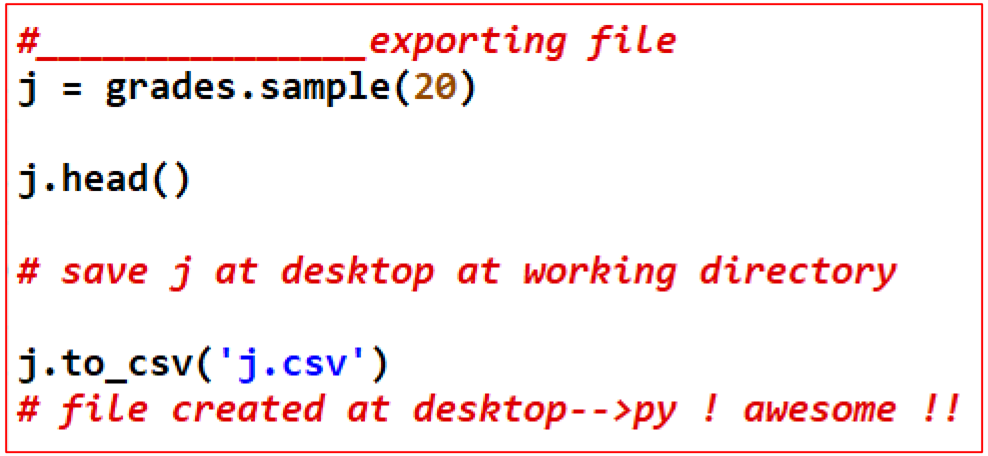


```
In [223]: #___________________________________cross tabulation

In [224]: # ethnicity versus gender

In [225]: pd.crosstab(grades.ethnicity, grades.gender, margins = True)
Out[225]:
gender      1   2  All
ethnicity
1           4   1    5
2          13   7   20
3          14  10   24
4          26  19   45
5           7   4   11
All        64  41  105

In [226]: # margins = True gives row column totals also
```

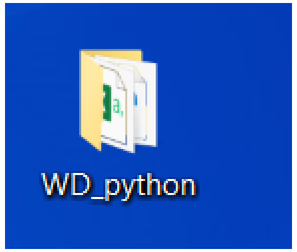
```
pd.crosstab(grades.ethnicity, grades.gender, margins = False)
# margins = False DO NOT give row column totals
```

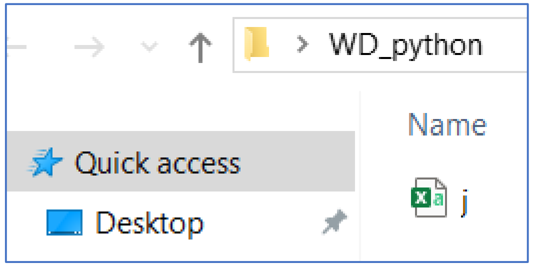


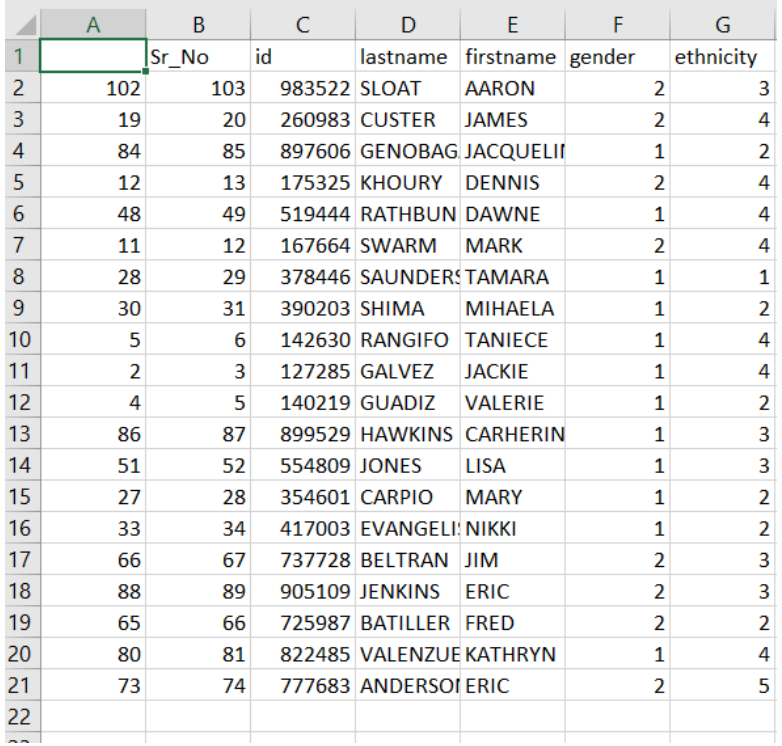
```
In [227]: pd.crosstab(grades.ethnicity, grades.gender, margins = False)
Out[227]:
gender      1   2
ethnicity
1           4   1
2          13   7
3          14  10
4          26  19
5           7   4

In [228]: # margins = False DO NOT give row column totals
```



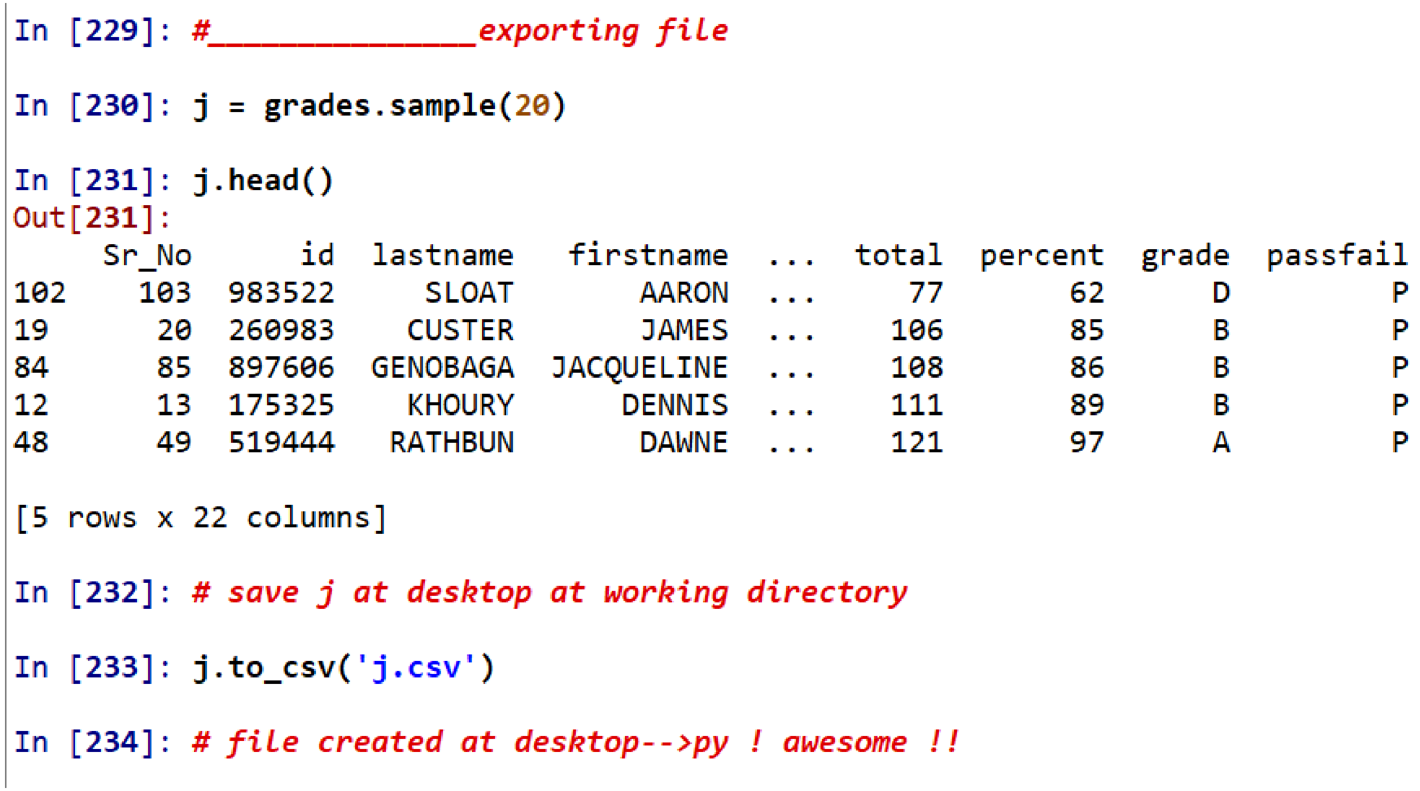
```
#_______________exporting file
j = grades.sample(20)

j.head()

# save j at desktop at working directory

j.to_csv('j.csv')
# file created at desktop-->py ! awesome !!
```







	A	B	C	D	E	F	G
1		Sr_No	id	lastname	firstname	gender	ethnicity
2	102	103	983522	SLOAT	AARON	2	3
3	19	20	260983	CUSTER	JAMES	2	4
4	84	85	897606	GENOBAG	JACQUELII	1	2
5	12	13	175325	KHOURY	DENNIS	2	4
6	48	49	519444	RATHBUN	DAWNE	1	4
7	11	12	167664	SWARM	MARK	2	4
8	28	29	378446	SAUNDERS	TAMARA	1	1
9	30	31	390203	SHIMA	MIHAELA	1	2
10	5	6	142630	RANGIFO	TANIECE	1	4
11	2	3	127285	GALVEZ	JACKIE	1	4
12	4	5	140219	GUADIZ	VALERIE	1	2
13	86	87	899529	HAWKINS	CARHERIN	1	3
14	51	52	554809	JONES	LISA	1	3
15	27	28	354601	CARPIO	MARY	1	2
16	33	34	417003	EVANGELI	NIKKI	1	2
17	66	67	737728	BELTRAN	JIM	2	3
18	88	89	905109	JENKINS	ERIC	2	3
19	65	66	725987	BATILLER	FRED	2	2
20	80	81	822485	VALENZUE	KATHRYN	1	4
21	73	74	777683	ANDERSOI	ERIC	2	5
22							

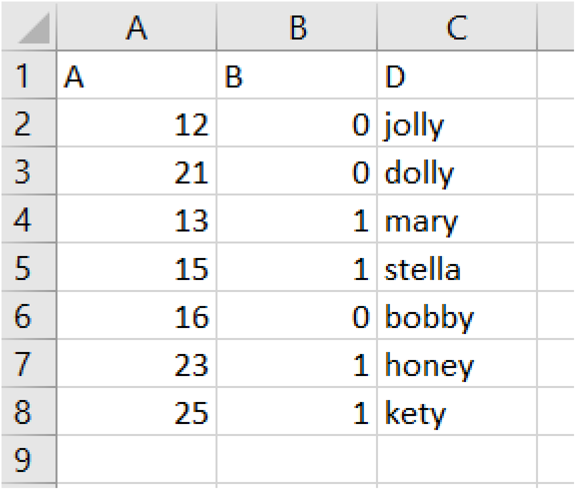


```
In [229]: #_______________exporting file

In [230]: j = grades.sample(20)

In [231]: j.head()
Out[231]:
    Sr_No      id  lastname   firstname  ...  total  percent  grade  passfail
102   103  983522     SLOAT       AARON  ...     77       62      D         P
19     20  260983    CUSTER       JAMES  ...    106       85      B         P
84     85  897606  GENOBAGA  JACQUELINE  ...    108       86      B         P
12     13  175325    KHOURY      DENNIS  ...    111       89      B         P
48     49  519444   RATHBUN       DAWNE  ...    121       97      A         P

[5 rows x 22 columns]

In [232]: # save j at desktop at working directory

In [233]: j.to_csv('j.csv')

In [234]: # file created at desktop-->py ! awesome !!
```

Task 1: All 0s (zeros) in column B to be changed to 2

Task 2: In column C, stella to be replaced by steffi

Task 3: Column names to be A as Marks, B as Section, D as Names



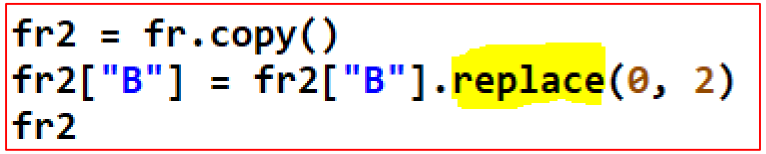
	A	B	C
1	A	B	D
2	12	0	jolly
3	21	0	dolly
4	13	1	mary
5	15	1	stella
6	16	0	bobby
7	23	1	honey
8	25	1	kety
9			

Task 1: All 0s (zeros) in column B to be changed to 2

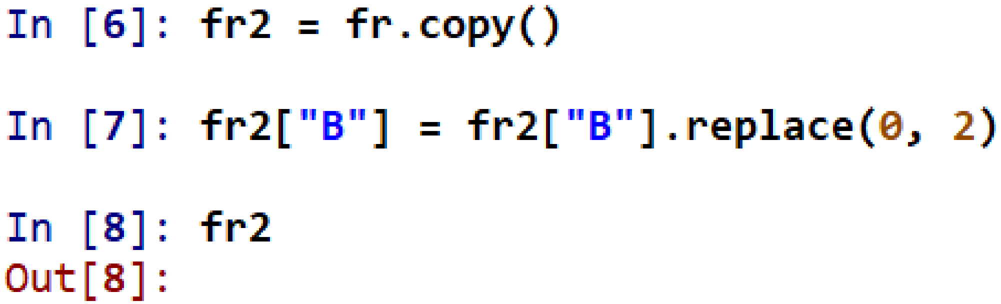
```
#___________________________________0 to 2
import pandas as pd
import numpy as np

fr = pd.read_csv("C:/Users/Dr Vinod/Desktop/DataSets1/FindReplace.csv")
fr = pd.DataFrame(fr)
fr
```

```
In [1]: import pandas as pd

In [2]: import numpy as np

In [3]: fr = pd.read_csv("C:/Users/Dr Vinod/Desktop/DataSets1/
FindReplace.csv")

In [4]: fr = pd.DataFrame(fr)

In [5]: fr
Out[5]:
    A  B       D
0  12  0   jolly
1  21  0   dolly
2  13  1    mary
3  15  1  stella
4  16  0   bobby
5  23  1   honey
6  25  1    kety
```

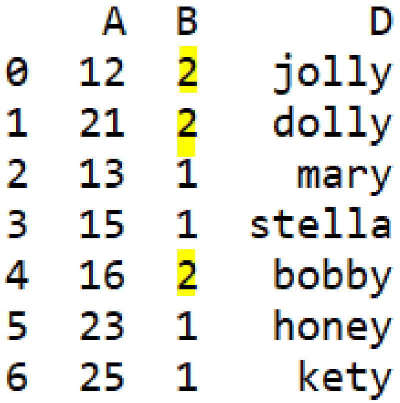
Task 1: All 0s (zeros) in column B to be changed to 2



```
fr2 = fr.copy()
fr2["B"] = fr2["B"].replace(0, 2)
fr2
```



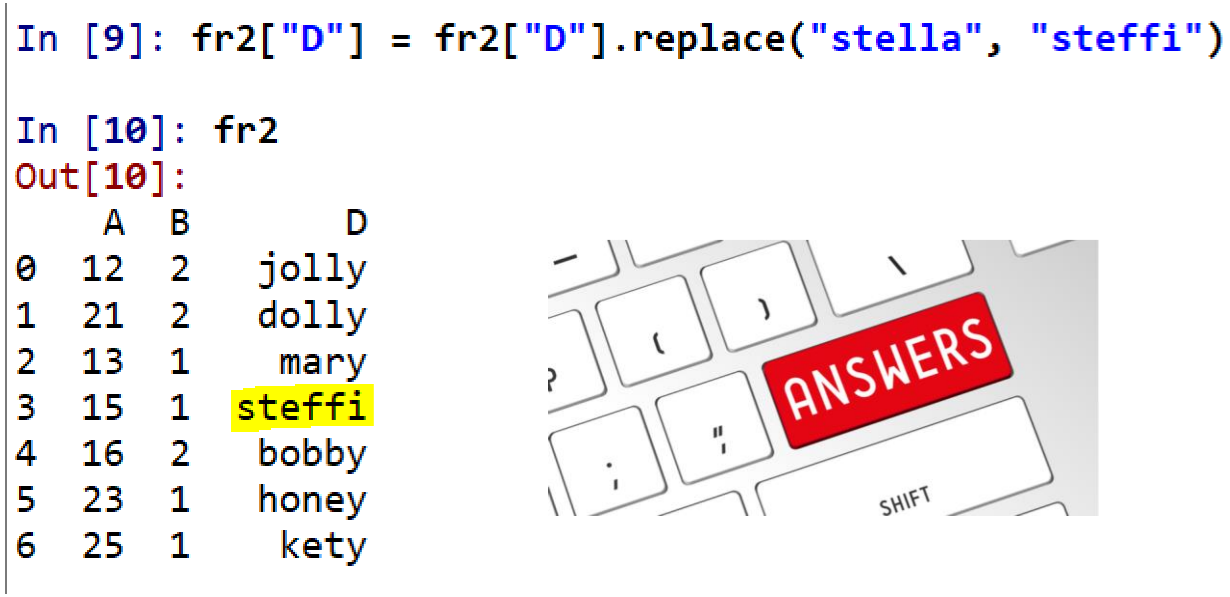
```
In [6]: fr2 = fr.copy()

In [7]: fr2["B"] = fr2["B"].replace(0, 2)

In [8]: fr2
Out[8]:
```



```
    A  B       D
0  12  2   jolly
1  21  2   dolly
2  13  1    mary
3  15  1  stella
4  16  2   bobby
5  23  1   honey
6  25  1    kety
```

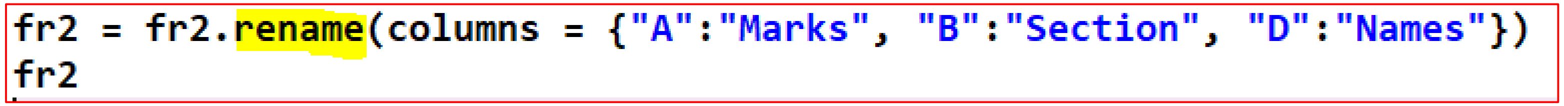
Task 2: In column C, stella to be replaced by steffi

```
fr2["D"] = fr2["D"].replace("stella", "steffi")
fr2
```

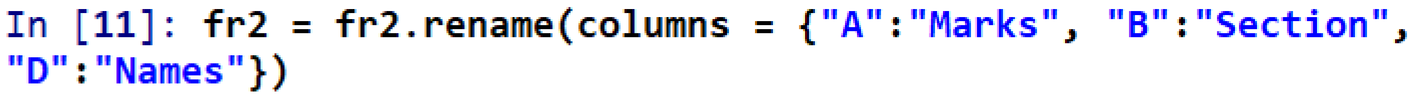


```
In [9]: fr2["D"] = fr2["D"].replace("stella", "steffi")

In [10]: fr2
Out[10]:
    A  B       D
0  12  2   jolly
1  21  2   dolly
2  13  1    mary
3  15  1  steffi
4  16  2   bobby
5  23  1   honey
6  25  1    kety
```

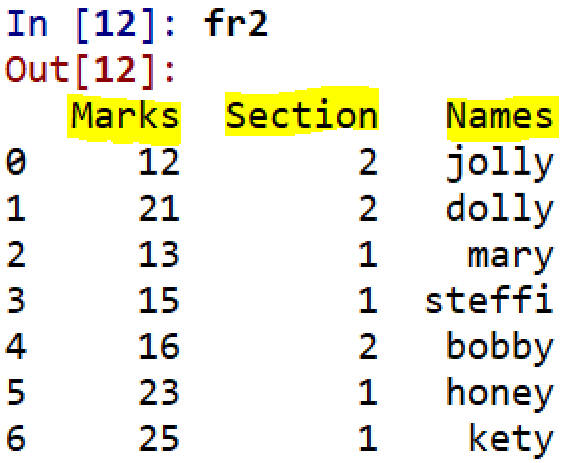
Task 3: Column names to be A as Marks, B as Section, D as Names



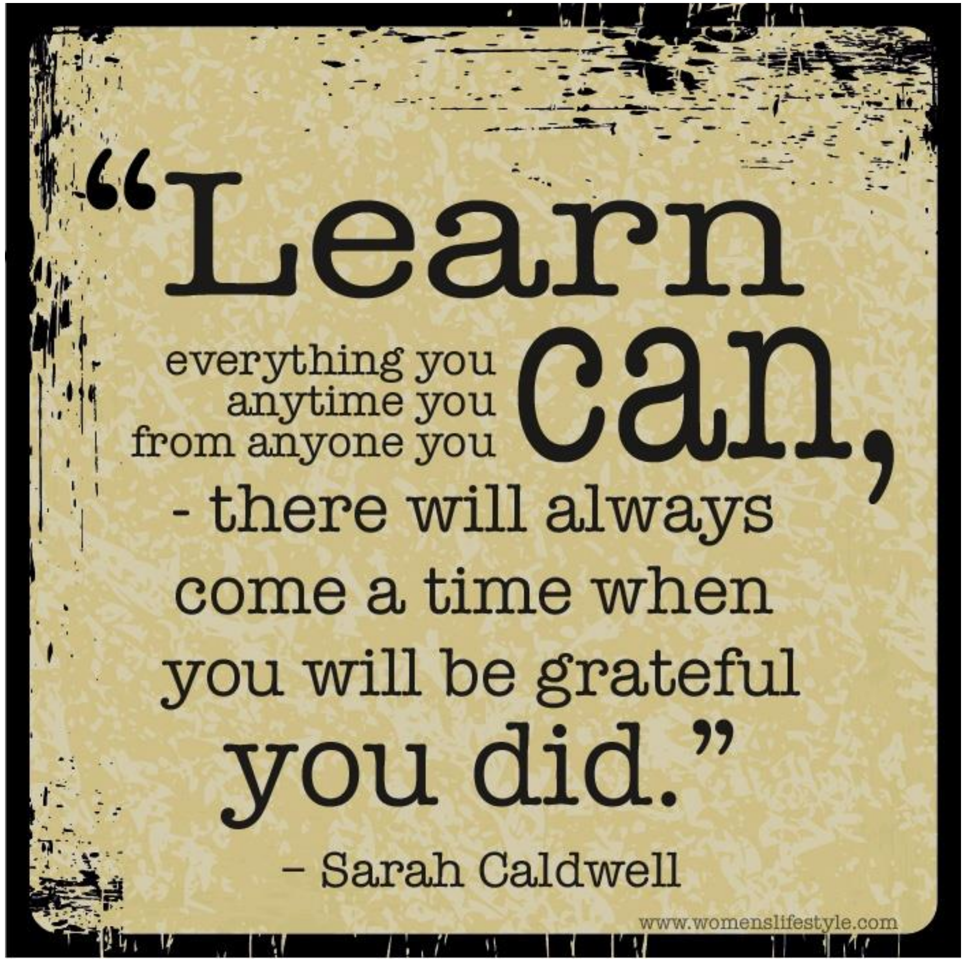
```
fr2 = fr2.rename(columns = {"A":"Marks", "B":"Section", "D":"Names"})
fr2
```



```
In [11]: fr2 = fr2.rename(columns = {"A":"Marks", "B":"Section",
"D":"Names"})
```



```
In [12]: fr2
Out[12]:
   Marks  Section   Names
0     12        2   jolly
1     21        2   dolly
2     13        1    mary
3     15        1  steffi
4     16        2   bobby
5     23        1   honey
6     25        1    kety
```



"Learn everything you can, anytime you can, from anyone you can, - there will always come a time when you will be grateful you did."

– Sarah Caldwell

www.womenslifestyle.com